Lecture 1: Problems and solutions. Optimality conditions for unconstrained optimization

Coralia Cartis, Mathematical Institute, University of Oxford

C6.2/B2: Continuous Optimization

Problems and solutions

$$\text{minimize } f(x) \quad \text{subject to } x \in \Omega \subseteq \mathbb{R}^n. \qquad (\dagger)$$

- $f : \Omega \to \mathbb{R}$ is (sufficiently) smooth ($f \in \mathcal{C}^i(\Omega)$, $i \in \{1, 2\}$).
- f objective; x variables; Ω feasible set (determined by finitely many constraints).
- n may be large.
- minimizing $-f(x) \equiv -$ maximizing $f(x)$. Wlog, minimize.

Problems and solutions

minimize $f(x)$ subject to $x \in \Omega \subseteq \mathbb{R}^n$. (†)

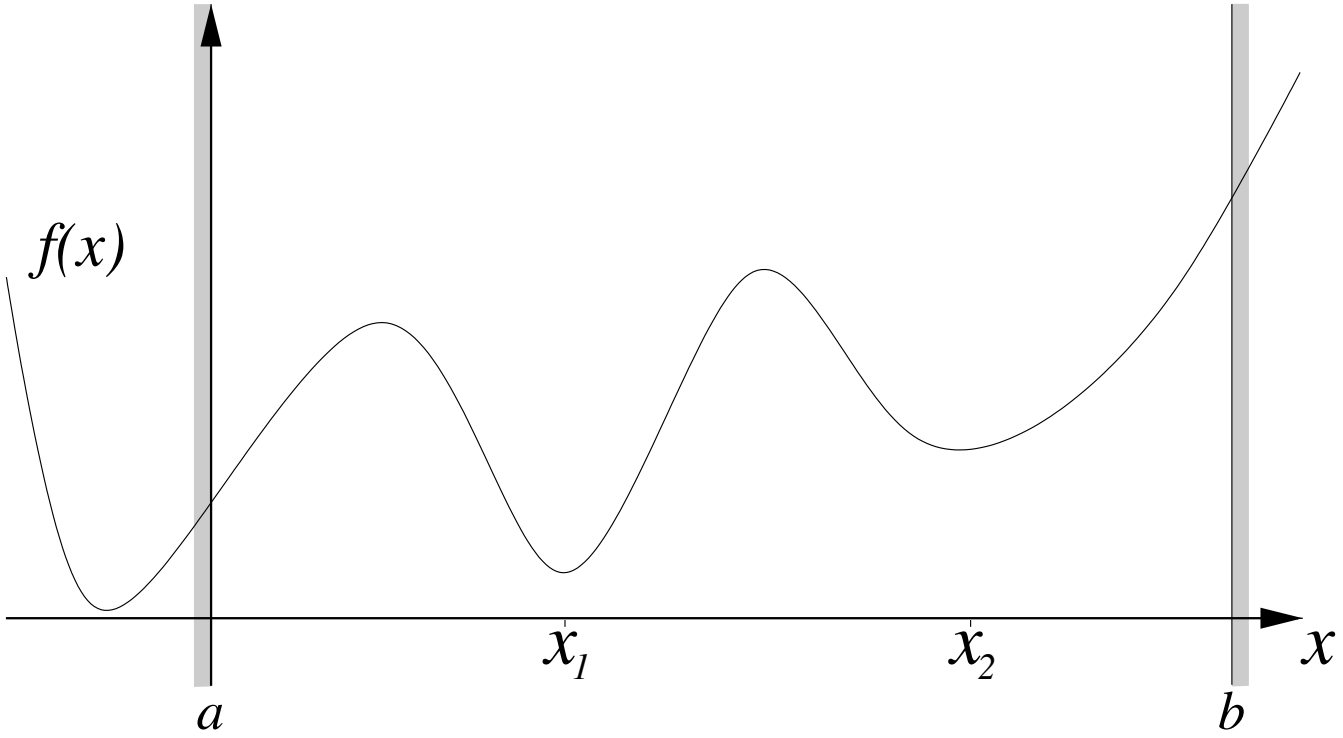
- $f : \Omega \to \mathbb{R}$ is (sufficiently) smooth ($f \in \mathcal{C}^i(\Omega)$, $i \in \{1, 2\}$).
- f objective; x variables; Ω feasible set (determined by finitely many constraints).
- n may be large.
- minimizing $-f(x) \equiv -$ maximizing $f(x)$. Wlog, minimize.

x^* global minimizer of f over $\Omega \iff f(x) \geq f(x^*)$, $\forall x \in \Omega$.

x^* local minimizer of f over $\Omega \iff$ there exists $\mathcal{N}(x^*, \delta)$ such that $f(x) \geq f(x^*)$, for all $x \in \Omega \cap \mathcal{N}(x^*, \delta)$,
where $\mathcal{N}(x^*, \delta) := \{x \in \mathbb{R}^n : \|x - x^*\| \leq \delta\}$ and $\|\cdot\|$ is the Euclidean norm.

Example problem in one dimension

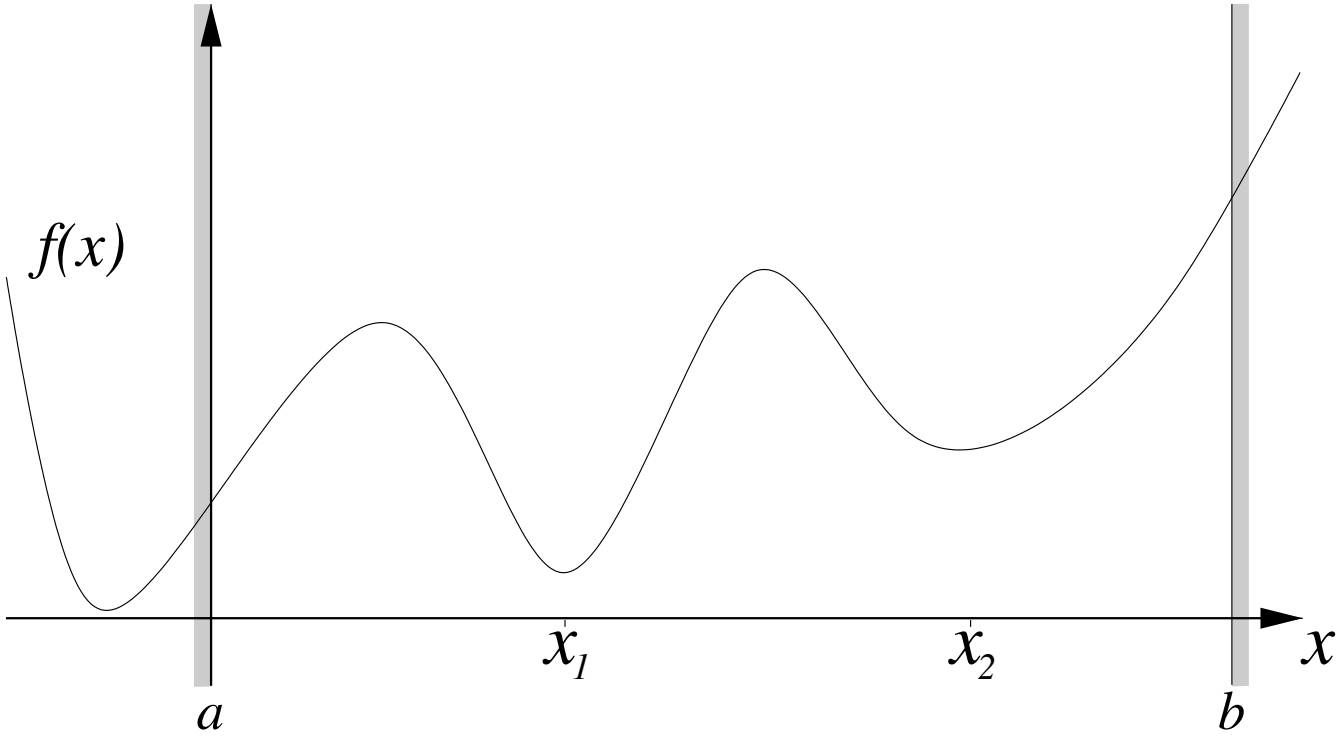
$$\textbf{Example :} \quad \min f(x) \quad \textbf{subject to} \quad a \leq x \leq b.$$



- The feasible region Ω is the interval $[a, b]$.

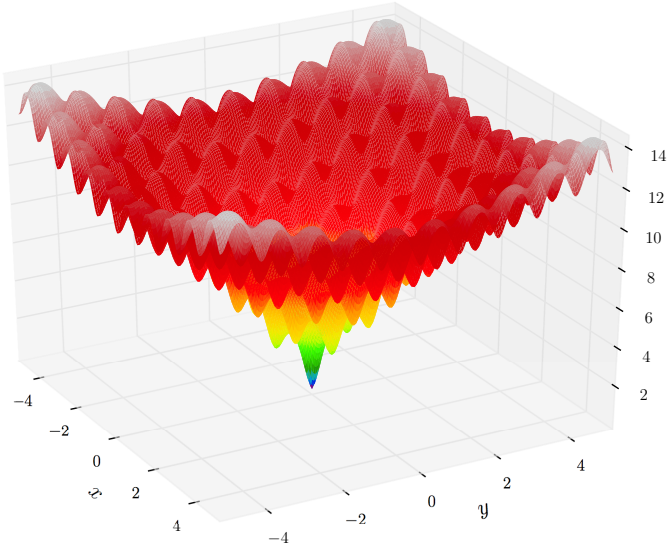
Example problem in one dimension

$$\textbf{Example :} \quad \min f(x) \quad \textbf{subject to} \quad a \leq x \leq b.$$

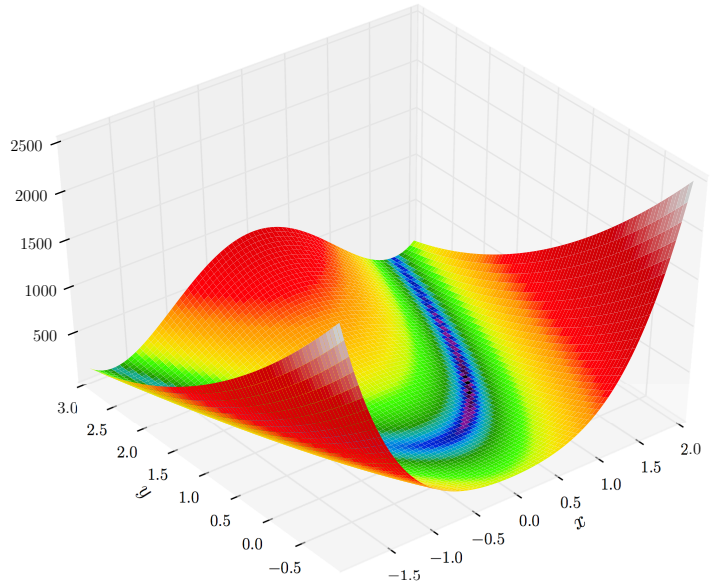


- The feasible region Ω is the interval $[a, b]$.
- The point x_1 is the global minimizer; x_2 is a local (non-global) minimizer; $x = a$ is a constrained local minimizer.

Example problems in two dimensions



Ackeley's test function



Rosenbrock's test function

[see Wikipedia]

Main classes of continuous optimization problems

Linear (Quadratic) programming: linear (quadratic) objective and linear constraints in the variables

$$\min_{x \in \mathbb{R}^n} c^T x \left(+ \frac{1}{2} x^T H x \right) \text{ subject to } a_i^T x = b_i, i \in E; \; a_i^T x \geq b_i, i \in I,$$

where c, $a_i \in \mathbb{R}^n$ for all i and H is $n \times n$ matrix; E and I are finite index sets.

Main classes of continuous optimization problems

Linear (Quadratic) programming: linear (quadratic) objective and linear constraints in the variables

$$\min_{x \in \mathbb{R}^n} c^T x \left(+ \frac{1}{2} x^T H x \right) \text{ subject to } a_i^T x = b_i, i \in E; \; a_i^T x \geq b_i, i \in I,$$

where c, $a_i \in \mathbb{R}^n$ for all i and H is $n \times n$ matrix; E and I are finite index sets.

Unconstrained (Constrained) nonlinear programming

$$\min_{x \in \mathbb{R}^n} f(x) \; (\text{subject to } c_i(x) = 0, i \in E; \; c_i(x) \geq 0, i \in I)$$

where $f, c_i : \mathbb{R}^n \longrightarrow \mathbb{R}$ are (smooth, possibly nonlinear) functions for all i; E and I are finite index sets.
Most real-life problems are nonlinear, often large-scale !

Example: an OR application [Gould'06]

Optimization of a high-pressure gas network
pressures $p = (p_i,\ i)$; flows $q = (q_j,\ j)$; demands $d = (d_k,\ k)$; compressors. Maximize net flow s.t. the constraints:

$$\begin{cases} Aq - d = 0 \\ A^T p^2 + K q^{2.8359} = 0 \\ A_2^T q + z \cdot c(p,q) = 0 \\ p_{\min} \le p \le p_{\max} \\ q_{\min} \le q \le q_{\max} \end{cases}$$

- $A,\ A_2 \in \{\pm 1, 0\};\ z \in \{0, 1\}$
- 200 nodes and pipes, 26 machines: 400 variables;
- variable demand, (p, d) 10mins. $\longrightarrow$ 58,000 vars; real-time.

Example: an inverse problem application[MetOffice]

Data assimilation for weather forecasting

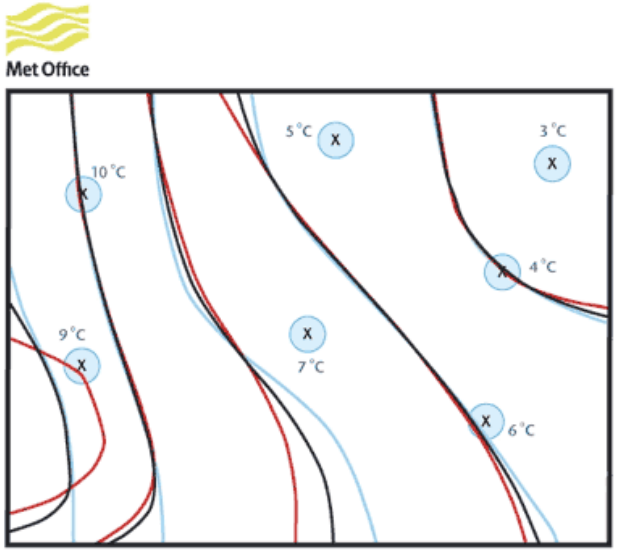
- best estimate of the current state of the atmosphere

$\longrightarrow$ find initial conditions x_0 for the numerical forecast by solving the (ill-posed) nonlinear inverse problem

$$\min_{x_0} \sum_{i=0}^{m} (H_i[x_i] - y_i)^T R_i^{-1} (H[x_i] - y_i),$$

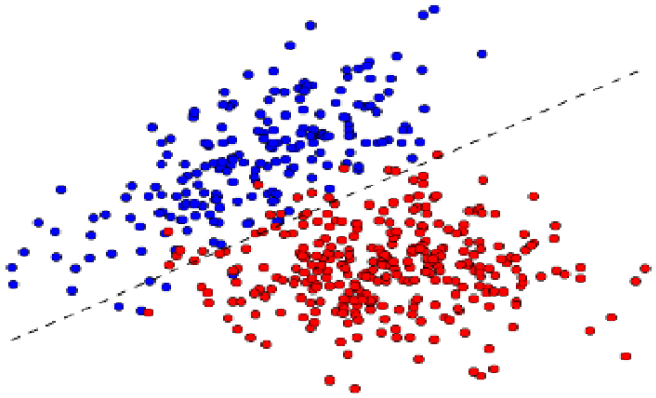
$x_i = S(t_i, t_0, x_0)$, S solution operator of the discrete nonlinear model; H_i maps x_i to observations y_i, R_i error covariance matrix of the observations at t_i

x_0 of size $10^7 - 10^8$;
observations $m \approx 250{,}000$.



Supervised learning problems

[Scheinberg, 2018; Curtis & Scheinberg, 2017; Bouttou et al, 2018]



Binary classification: Map $w \in \mathcal{W} \subseteq \Re^{d_w}$ to $y \in \mathcal{Y} \subseteq \{-1, 1\}$

Choose predictor $p(w; x) : \mathcal{W} \to \mathcal{Y}$

If $p(w; x) = w^T x$ - linear classifier; more generally, $p(w; x)$ nonlinear (such as neural network).

Selection of the best classifier:

- Minimize Expected/Empirical Error, Loss, AUC

Finding the best predictor

[Curtis & Scheinberg, 2017; Scheinberg, 2018]

$$\min_{x \in \mathcal{X}} f(x) := \int_{\mathcal{W} \times \mathcal{Y}} 1[yp(w; x) \leq 0] dP(w, y).$$

$\longrightarrow$ intractable due to unknown distribution

Use instead the empirical risk of $p(w; x)$ over finite training set $\mathcal{S}$,

$$\min_{x \in \mathcal{X}} f_{\mathcal{S}}(x) := \frac{1}{m} \sum_{i=1}^{m} 1[y_i p(w_i; x) \leq 0].$$

$\longrightarrow$ hard to solve, nonsmooth.

Use the smooth and 'easy' empirical loss of $p(w; x)$ over the finite training set $\mathcal{S}$,

$$\min_{x \in \mathcal{X}} \hat{f}_{\mathcal{S}}(x) := \frac{1}{m} \sum_{i=1}^{m} l(p(w_i; x), y_i) = \sum_{i=1}^{m} f_i(x).$$

$\longrightarrow$ tractable but huge scale in n and m; deterministic formulation.

Care also about expected loss $\mathbb{E}(l(p(w; x), y))$ (stochastic).

Lecture Course Outline

- Unconstrained optimization :
  - optimality conditions (characterizing optimality solutions of unconstrained problems)
  - algorithms/methods : gradient and Newton methods, line-search and trust-region techniques to ensure convergence.
- Constrained optimization :
  - optimality conditions (characterizing optimality solutions of constrained problems)
  - algorithms/methods : penalty methods, augmented Lagrangian, interior point algorithms and time permitting, more.

Problem/intercollegiate classes: 4.
Numerical laboratories: 2 (optional, details TBC.)

C6.2/B2. Continuous Optimization

Resources

References

[1] A. R. Conn, N. I. M. Gould and Ph. L. Toint, *Trust-Region Methods*, SIAM 2000.

[2] J. Dennis and R. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear equations*, (republished by) SIAM, 1996.

[3] R. Fletcher, *Practical Methods of Optimization*, 2nd edition Wiley, 1987 (republished in paperback in 2000).

[4] P. Gill, W. Murray and M. H. Wright, *Practical Optimization*, Academic Press, 1981.

on course

[5] N. I. M. Gould, *An Introduction to Algorithms for Continuous Optimization*, 2006. Available for download at `http://www.numerical.rl.ac.uk/nimg/course/lectures/paper/paper.pdf`.

[6] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Verlag, 1999 (1st edition) or 2006 (2nd edition). All citations in the lecture notes apply to either edition, unless otherwise stated.

Comments on the bibliography

For a comprehensive, yet highly accessible, introduction to numerical methods for continuous (unconstrained and constrained) optimization problems, see [6] - most recommended (but not required) for this course ! Reference [5] is also a very good, but more succinct introduction to this topic, with particular emphasis on nonconvex problems and with a well-structured bibliography of fundamental optimization articles. The monograph [1] is the most comprehensive reference book on trust-region methods to date. The remaining books in the bibliography are classics of the nonlinear (constrained and unconstrained) optimization literature.

Online and software resources

For an index and a guide to existing public and commercial software for solving (constrained and unconstrained) optimization problems, see

`http://neos-guide.org/Optimization-Guide`

and follow the links to *Optimization Tree* for example. Other useful links related to optimization may be found at the same webpage (links to test problems, to the NEOS Server which solves user-sent optimization problems over the internet, to online repositories of optimization articles, etc.).

For general nonconvex, smooth constrained and unconstrained problems the following software packages are of high quality/reliable: KNITRO, IPOPT, GALAHAD, etc. MATLAB's optimization toolbox (available on departmental computers) contains built-in optimization solvers for various problem classes - be careful which subroutine you choose ! COIN-OR is a public software repository that you may find useful in the future.

An important aspect of optimization software is the *interface* that allows the user to input the problem to the solver; interfaces, and hence acceptable input formats, vary between solvers. Presently, usually besides file-input in the language the solver is written in, much software allows MATLAB input files or/and AMPL files (AMPL is a modelling language specifically designed for expressing optimization problems; see `www.ampl.com`), etc.

An introduction to algorithms for continuous optimization

AVAILABLE ON COURSE WEBSITE

COURSE SLIDES BASED OFTEN ON THIS BOOKLET

Nicholas Gould

Oxford University Computing Laboratory
and
Rutherford Appleton Laboratory

Contents

C6.2/B2. Continuous Optimization

Mathematical Background (brief review)

Optimization draws on a number of key results in analysis and linear algebra. We briefly summarize some useful notions here. For more details, you may consult **Burden, R.L., & Faires, J.D.**, *Numerical Analysis*, 6th edition or later, Brooks/Cole Publishing.

Single valued functions and their derivatives

All the functions $f : \mathbb{R}^n \mapsto \mathbb{R}$ in this course are assumed to be smooth.

- The function $l : \mathbb{R}^n \mapsto \mathbb{R}$ is a **linear function** iff it is of the form

$$l(x) = d + g^T x \equiv d + \sum_{i=1}^{n} g_i x_i, \quad \text{where} \quad g = \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix},$$

  and $d \in \mathbb{R}$ and $g \in \mathbb{R}^n$ are known.

- The function $q(x) : \mathbb{R}^n \mapsto \mathbb{R}$ is a **quadratic function** iff it is of the form

$$q(x) = d + g^T x + \frac{1}{2} x^T H x = d + \sum_{i=1}^{n} g_i x_i + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} h_{ij} x_i x_j, \text{ where } H = \begin{pmatrix} h_{11} & h_{12} & \dots & h_{1n} \\ h_{21} & h_{22} & \dots & h_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n1} & h_{n2} & \dots & h_{nn} \end{pmatrix}.$$

  may be taken to be constant and symmetric. Although a quadratic function is strictly nonlinear, its properties are such that it is treated separately. Thus the term 'nonlinear function' often refers to a function which is not linear *or* quadratic.

- For the function $f : \mathbb{R}^n \mapsto \mathbb{R}$, the **vector of first partial derivatives** or **gradient vector** is

$$g(x) \equiv \nabla f(x) = \begin{pmatrix} \dfrac{\partial f}{\partial x_1} \\ \dfrac{\partial f}{\partial x_2} \\ \vdots \\ \dfrac{\partial f}{\partial x_n} \end{pmatrix}(x),$$

  where ∇ denotes the gradient operator $(\partial/\partial x_1 \;\; \partial/\partial x_2 \;\; \dots \;\; \partial/\partial x_n)^T$.

- For the function $f : \mathbb{R}^n \mapsto \mathbb{R}$, the **matrix of second partial derivatives** or **Hessian matrix**

$$H(x) \equiv \nabla [g(x)]^T = \nabla [\nabla f(x)]^T = \nabla \nabla^T f(x) = \nabla^2 f(x),$$