Convex Optimization: Geometric Programming

course project by

DMITRII GUREV
M03-201

Optimization in Machine Learning,
Fall 2022

Table of contents

Geometric Programming

Examples

Geometric Programming

Monomials and posynomials

For $x \in \mathbf{R}^n_{++}$

Monomial: $$f(x) = c x_1^{a_1} x_2^{a_2} \dots x_n^{a_n} \quad \text{with } c > 0$$

Posynomial: $$f(x) = \sum_{k=1}^{K} c_k x_1^{a_{1k}} x_2^{a_{2k}} \dots x_n^{a_{nk}} \quad \text{with } c_k > 0$$

Geometric program

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \le 1, \quad i = 1, \ldots, m \\ & h_i(x) = 1, \quad i = 1, \ldots, p, \end{array}$$

where $f_0, \ldots, f_m$ are posynomials and $h_1, \ldots, h_p$ are monomials

(the constraint $x \succ 0$ is implicit)

Convex optimization problem

$$
\begin{array}{ll}
\text{minimize} & f_0(x) \\
\text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\
& a_i^T x = b_i, \quad i = 1, \ldots, p
\end{array}
$$

where $f_0, \ldots, f_m$ are convex functions

Geometric program in convex form

Change of variables: $y_i = \log x_i$

Monomial:
$$\begin{aligned} f(x) &= c\left(e^{y_1}\right)^{a_1}\left(e^{y_2}\right)^{a_2}\ldots\left(e^{y_n}\right)^{a_n} \\ &= e^{a^T y + b}, \qquad b = \log c \end{aligned}$$

Posynomial:
$$f(x) = \sum_{k=1}^{K} e^{a_k^T y + b_k}$$

Geometric program in convex form

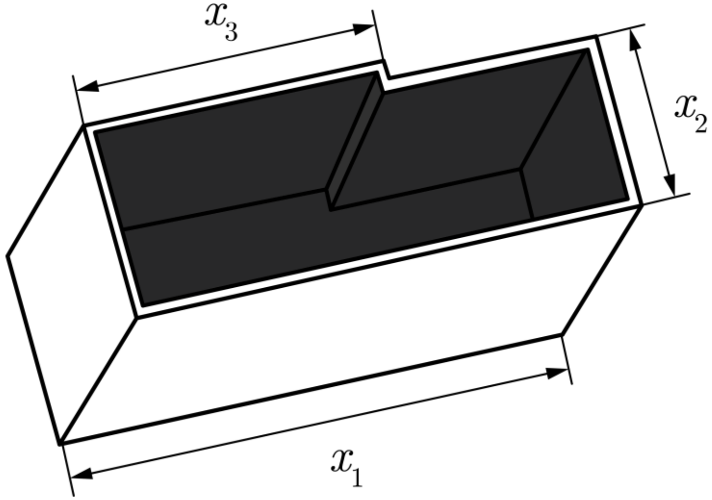
$$
\begin{array}{ll}
\text{minimize} & \displaystyle\sum_{k=1}^{K_0} e^{a_{0k}^T y + b_{0k}} \\
\text{subject to} & \displaystyle\sum_{k=1}^{K_i} e^{a_{ik}^T y + b_{ik}} \leq 1, \quad i = 1, \ldots, m \\
 & e^{g_i^T y + h_i} = 1, \quad i = 1, \ldots, p.
\end{array}
$$

Geometric program in convex form

$$
\begin{array}{ll}
\text{minimize} & \tilde{f}_0(y) = \log\left(\sum_{k=1}^{K_0} e^{a_{0k}^T y + b_{0k}}\right) \\
\text{subject to} & \tilde{f}_i(y) = \log\left(\sum_{k=1}^{K_i} e^{a_{ik}^T y + b_{ik}}\right) \leq 0, \quad i = 1, \ldots, m \\
& \tilde{h}_i(y) = g_i^T y + h_i = 0, \quad i = 1, \ldots, p.
\end{array}
$$

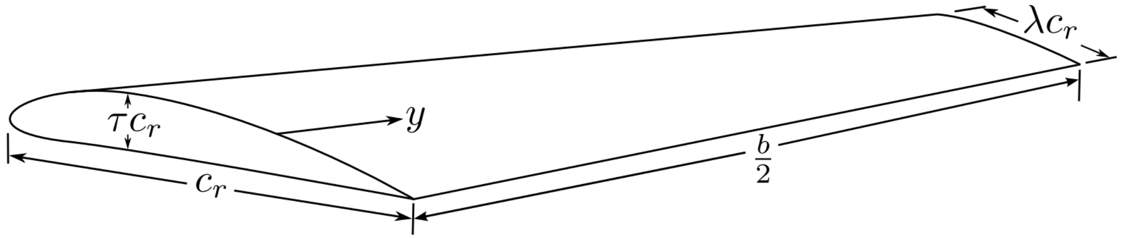
Applications

Component sizing



Huang, Chia-Hui. "Engineering design by geometric programming." *Mathematical problems in engineering* 2013 (2013).

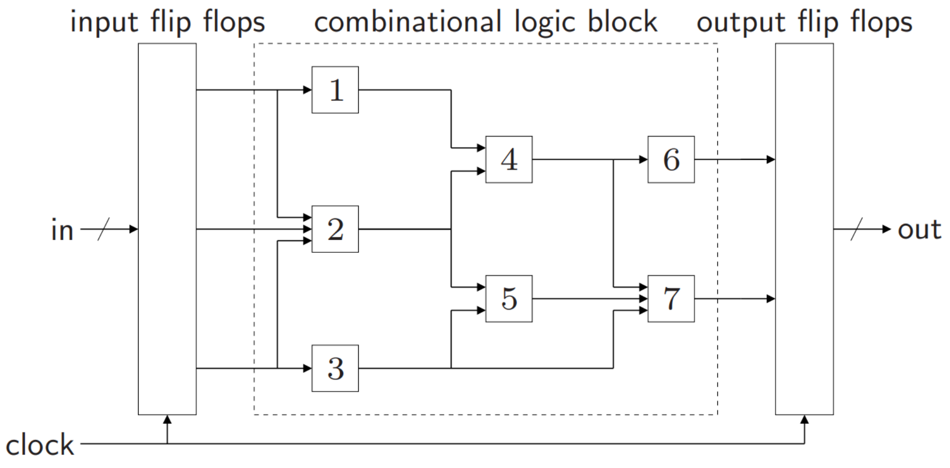
Aircraft design



Hoburg, Warren, and Pieter Abbeel. "Geometric programming for aircraft design optimization." *AIAA Journal* 52.11 (2014): 2414-2426.

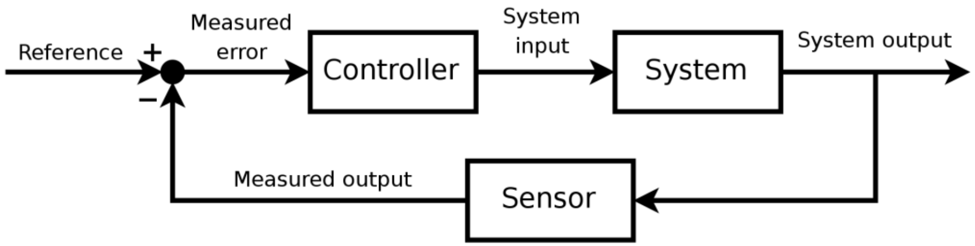
Applications

Electronic circuit design



Boyd, Stephen, Seung Jean Kim, and S. Mohan. "Geometric programming and its applications to EDA problems." *date tutorial* (2005).

Optimal control



Jefferson, T. R., and C. H. Scott. "Generalized geometric programming applied to problems of optimal control: I. Theory." *Journal of Optimization Theory and Applications* 26.1 (1978): 117-129.

Current state of the art

Interior-point methods (e.g. barrier method)

Approaching efficiency of linear programming solvers:

- 1,000 variables, 10,000 monomial terms: few seconds
- 10,000 variables, 100,000 monomial terms: minute
- 1,000,000 variables, 10,000,000 monomial terms: hour

these are order-of-magnitude estimates, on simple PC

Boyd, Stephen, Seung Jean Kim, and S. Mohan. "Geometric programming and its applications to EDA problems." *date tutorial* (2005).

Examples

Frobenius norm diagonal scaling

Linear function $y = Mu$

Scaling $\tilde{u} = Du \quad \tilde{y} = Dy \quad \tilde{y} = DMD^{-1}\tilde{u}$

D is a diagonal matrix with $D_{ii} = d_i > 0$

minimize $$||DMD^{-1}||_{\mathrm{F}}^2 = \mathbf{tr}\left((DMD^{-1})^T(DMD^{-1})\right) = \sum_{i,j=1}^{n}(DMD^{-1})_{ij}^2 = \sum_{i,j=1}^{n} M_{ij}^2 d_i^2/d_j^2.$$

Frobenius norm diagonal scaling

In convex form:

$$\text{minimize} \qquad \log\left(\sum_{i,j=1}^{n} e^{2x_i - 2x_j + \log M_{ij}^2}\right), \qquad x_i = \log d_i$$

Frobenius norm diagonal scaling

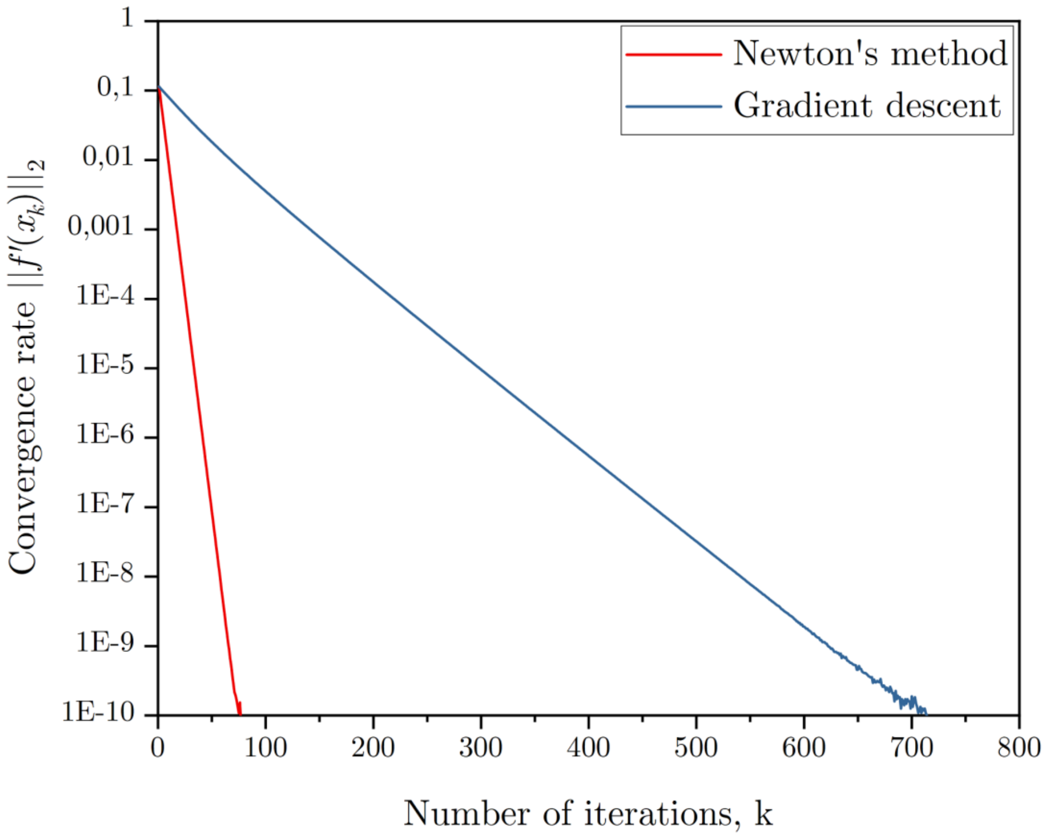
Gradient descent:

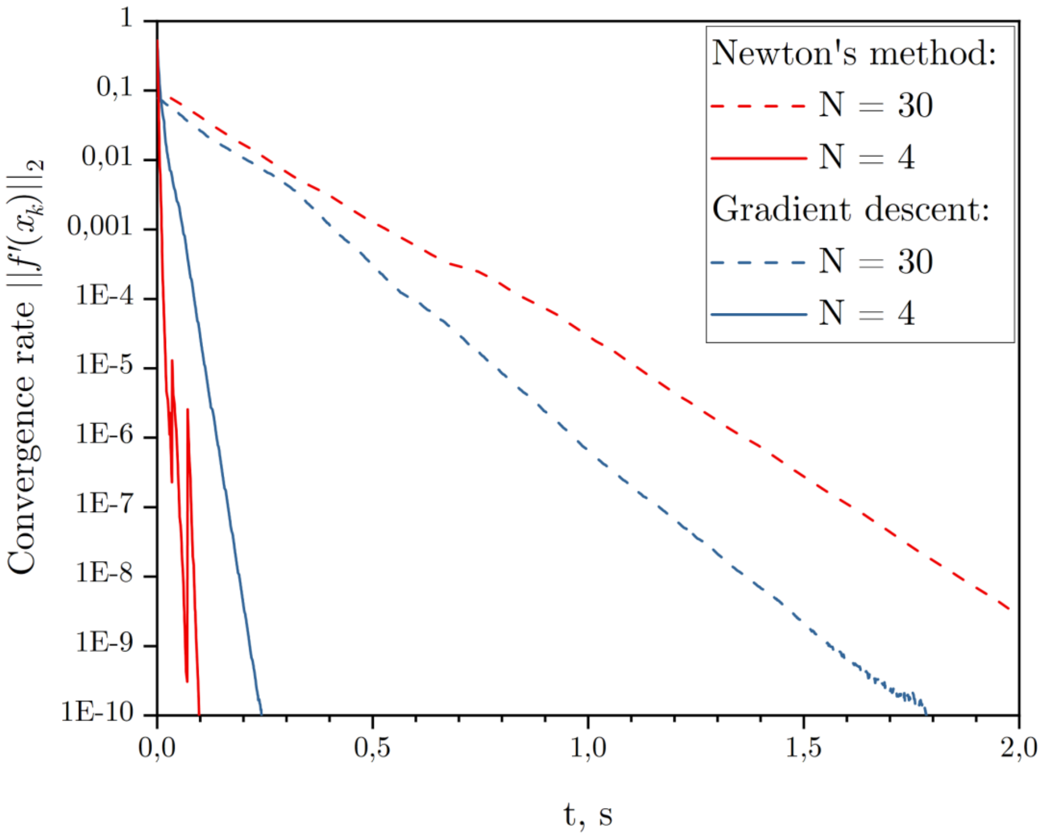
$$x_{k+1} = x_k - \alpha \nabla f(x_k), \quad \alpha = Const$$

Newton's method:

$$x_{k+1} = x_k - H(x_k)^{-1} \nabla f(x_k)$$

Frobenius norm diagonal scaling





Frobenius norm diagonal scaling

For $\quad M = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}$

the solution obtained is $\quad d = (0.529, \quad 0.373, \quad 0.314, \quad 0.283)^T$

$||M||_{\mathrm{F}} = 38,678 \qquad ||DMD^{-1}||_{\mathrm{F}} = 36,289$

Maximum area of a rectangle

$$\begin{aligned} \text{maximize} \quad & S(a,b) = ab \\ \text{subject to} \quad & a + b \leq \frac{p}{2}. \end{aligned}$$

The solution is well known: $a^* = \dfrac{p}{4}, \quad b^* = \dfrac{p}{4}$

Maximum area of a rectangle

In convex form:

$$\begin{aligned}
&\text{minimize} && -\log e^{A+B} = -(A+B) \\
&\text{subject to} && \log\left(e^{A+\log\frac{2}{p}} + e^{B+\log\frac{2}{p}}\right) \leq 0, \\
& && A = \log a,\ B = \log b
\end{aligned}$$

Maximum area of a rectangle

Use the barrier method to eliminate the inequality constraint:

$$\text{minimize} \qquad -A - B + I_{-}\left(\log\left[e^{A+\log\frac{2}{p}} + e^{B+\log\frac{2}{p}}\right]\right)$$

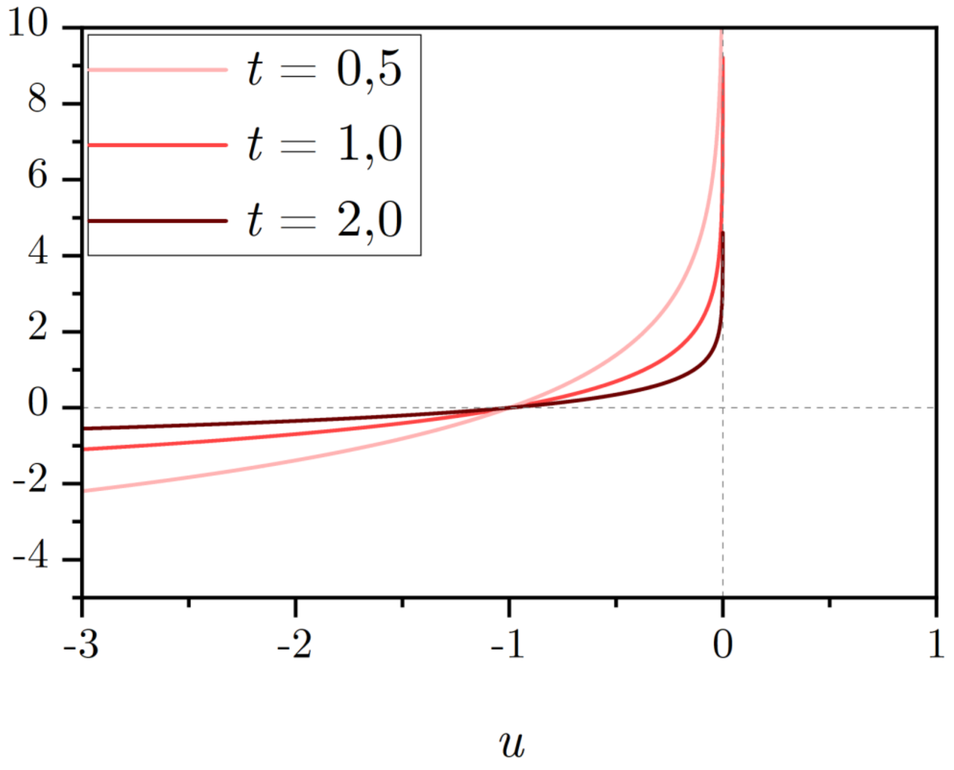
Indicator function $\quad I_{-}(u) = \begin{cases} 0, & u \leq 0 \\ \infty, & u > 0 \end{cases}$

Maximum area of a rectangle

Logarithmic barrier is a differentiable approximation of the indicator function

$$\hat{I}_{-}(u) = -\frac{1}{t}\log(-u),$$

$$\textbf{dom}\,\hat{I}_{-}(u) = -\mathbf{R}_{++}$$



Maximum area of a rectangle

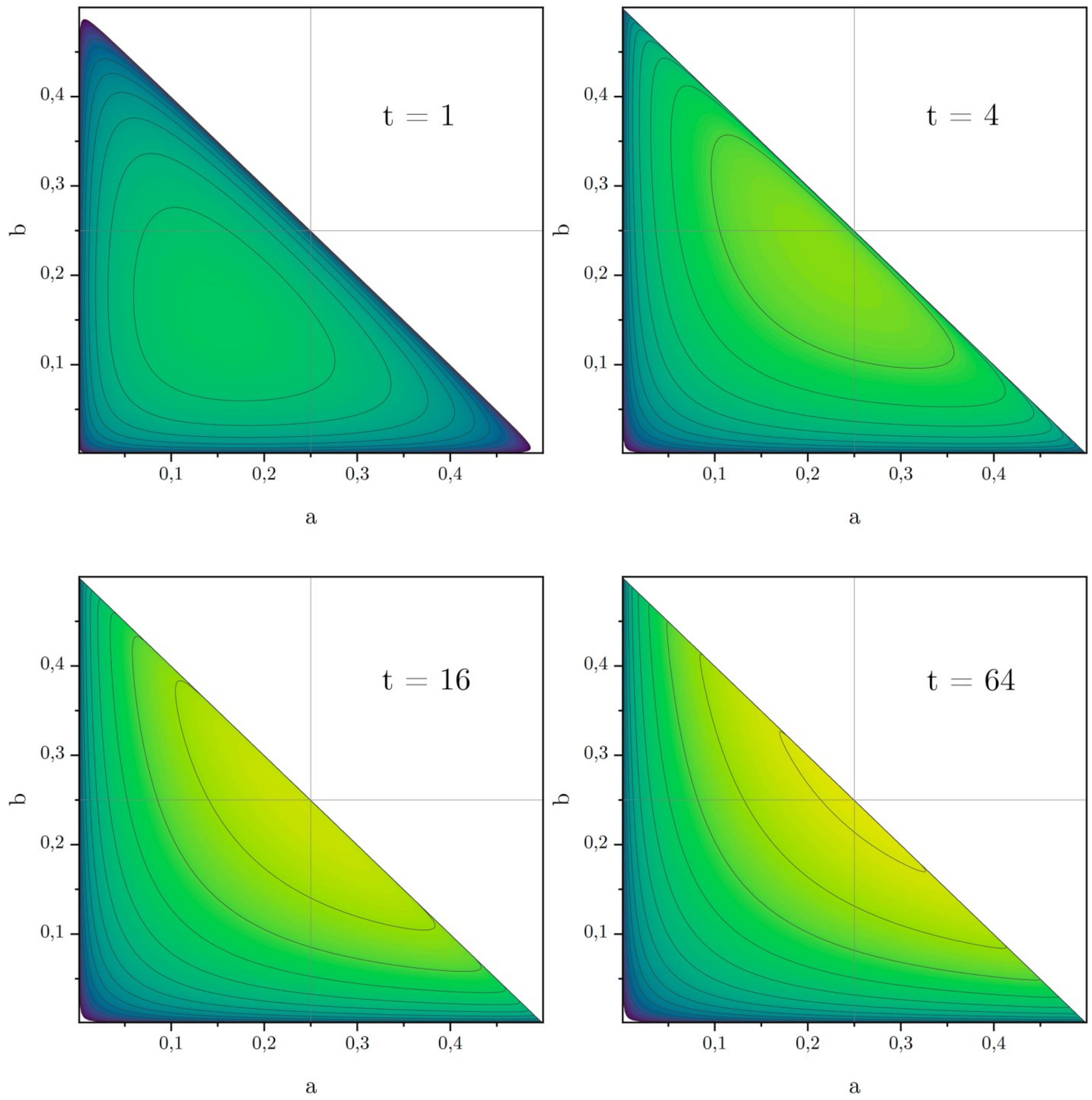
Now the objective is to

$$\text{minimize} \qquad -A - B - \frac{1}{t}\log\left(-\log\left[e^{A+\log\frac{2}{p}} + e^{B+\log\frac{2}{p}}\right]\right)$$

(merely an approximation of the original problem)

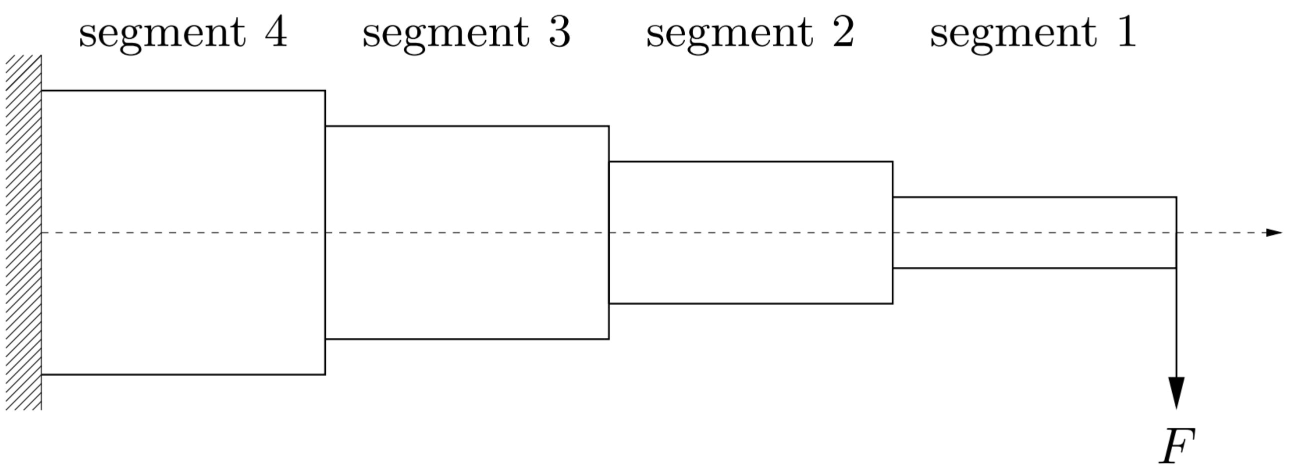
Maximum area of a rectangle

Solve a sequence of problems, increasing the parameter t (and therefore the accuracy of the approximation) at each step.



Design of a cantilever beam

N segments of a unit length, width w_i and height h_i



Minimize the volume $w_1 h_1 + \cdots + w_N h_N$

Design of a cantilever beam

Constraints:

$$w_{\min} \leq w_i \leq w_{\max}, \quad h_{\min} \leq h_i \leq h_{\max}, \quad i = 1, \ldots, N$$

$$\frac{6iF}{w_i h_i^2} \leq \sigma_{\max}, \quad i = 1, \ldots, N$$ maximum stress

$$y_1 \leq y_{\max}$$ maximum vertical deflection

Design of a cantilever beam

The deflection can be found recursively

$$v_i = 12(i - 1/2)\frac{F}{Ew_i h_i^3} + v_{i+1}$$ slope

$$y_i = 6(i - 1/3)\frac{F}{Ew_i h_i^3} + v_{i+1} + y_{i+1}$$ deflection

$$v_{N+1} = y_{N+1} = 0$$

Design of a cantilever beam

$$\begin{array}{ll}
\text{minimize} & \sum_{i=1}^{N} w_i h_i \\
\text{subject to} & w_{\min} \le w_i \le w_{\max}, \quad i = 1, \ldots, N \\
& h_{\min} \le h_i \le h_{\max}, \quad i = 1, \ldots, N \\
& 6iF/(w_i h_i^2) \le \sigma_{\max}, \quad i = 1, \ldots, N \\
& y_1 \le y_{\max}
\end{array}$$

For a 3-segment beam $y_1 = 4\frac{F}{Ew_1h_1^3} + 28\frac{F}{Ew_2h_2^3} + 76\frac{F}{Ew_3h_3^3}$

The convex form can be derived analogously to the previous examples

Design of a cantilever beam

Suppose $N = 3$

$$w_{\min} = h_{\min} = 0.1 \text{ m}$$

$$w_{\max} = h_{\max} = 0.5 \text{ m}$$

$$F = 3 \cdot 10^5 \text{ N}$$

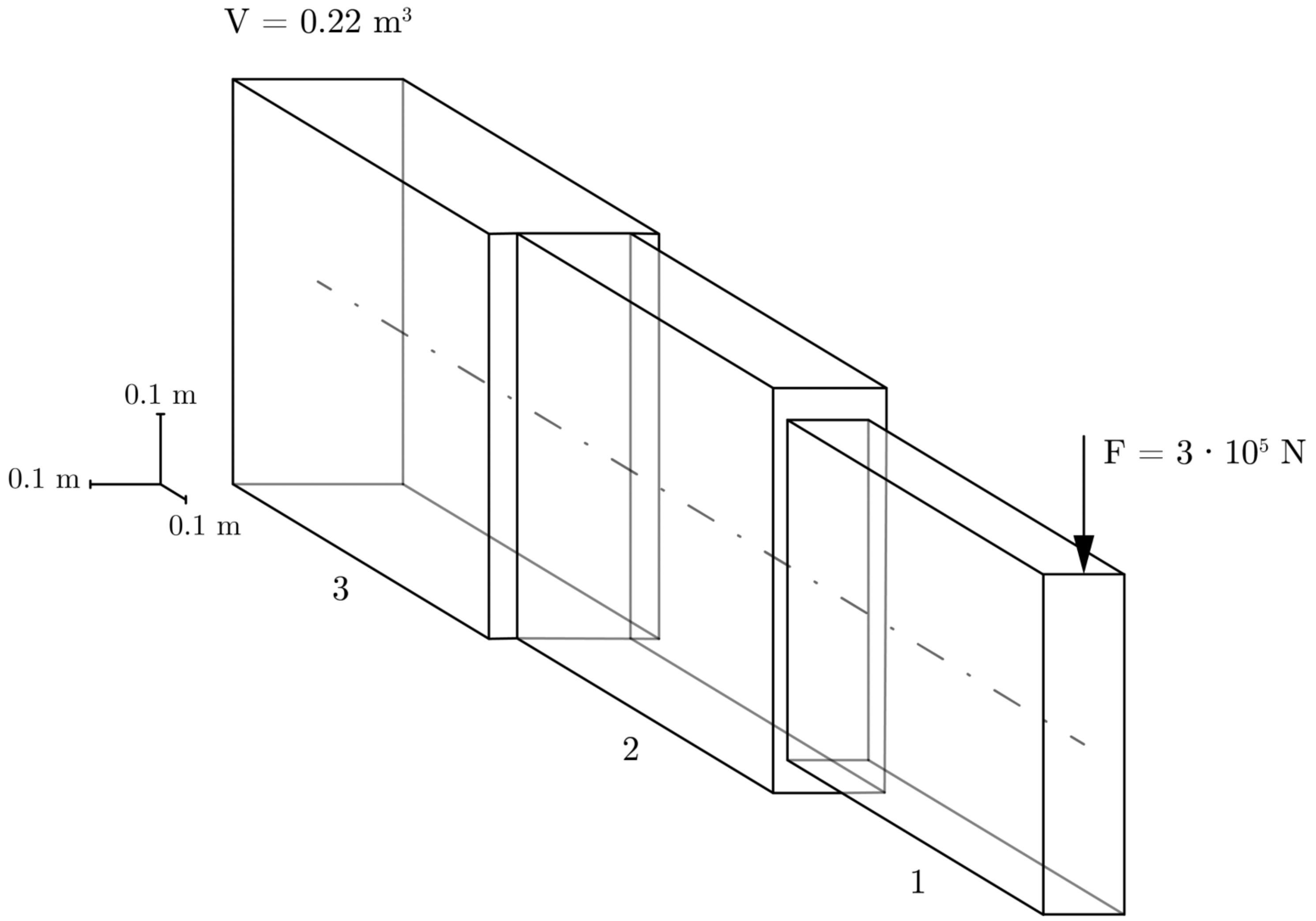
$$E = 2 \cdot 10^{11} \text{ Pa} \qquad \text{(steel)}$$

$$\sigma_{\max} = 10^8 \text{ Pa} \qquad \text{(steel)}$$

$$y_{\max} = 0.01 \text{ m}$$

Design of a cantilever beam

The optimal volume is $0.22\ \mathrm{m}^3$



Conclusions

Conclusions

Geometric programming

- comes up in a variety of contexts

- can be transformed to convex problems by a change of variables and a transformation of the objective and constraint functions

- admits fast, reliable solution of large-scale problems

Conclusions

Using Newton's method with the barrier method for eliminating inequality constraints, the following illustrative problems have been solved:

- Frobenius norm diagonal scaling

- Maximum area of a rectangle

- Design of a cantilever beam