



WIKIPEDIA  
The Free Encyclopedia

Main page  
Contents  
Featured content  
Current events  
Random article  
Donate to Wikipedia  
Wikipedia store

Interaction  
Help  
About Wikipedia  
Community portal  
Recent changes  
Contact page

Tools  
What links here  
Related changes  
Upload file  
Special pages  
Permanent link  
Page information  
Wikidata item  
Cite this page

Print/export  
Create a book  
Download as PDF  
Printable version

Languages  
Deutsch  
Français  
Русский [Edit links](#)

[Create account](#) [Log in](#)

Article [Talk](#)

[Read](#) [Edit](#) [View history](#)

Search

# Ellipsoid method

From Wikipedia, the free encyclopedia

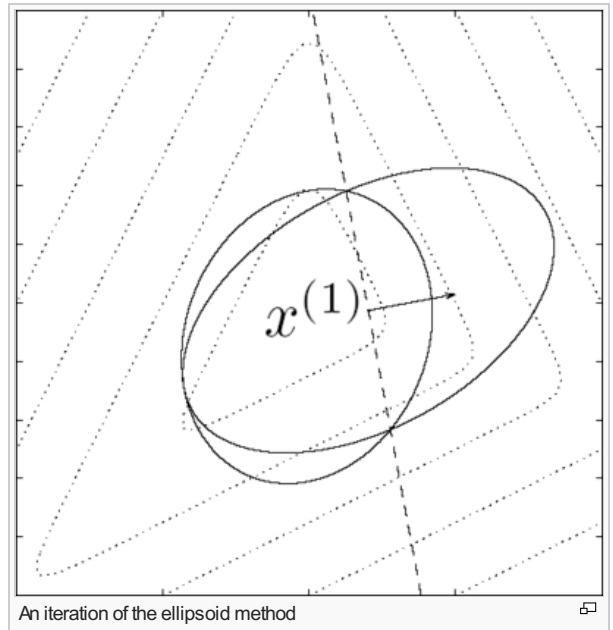
In [mathematical optimization](#), the **ellipsoid method** is an [iterative method](#) for [minimizing convex functions](#).

When specialized to solving feasible [linear optimization](#) problems with rational data, the ellipsoid method is an [algorithm](#) which finds an optimal solution in a finite number of steps.

The ellipsoid method generates a sequence of [ellipsoids](#) whose volume uniformly decreases at every step, thus enclosing a minimizer of a [convex function](#).

## Contents

- 1 History
- 2 Description
- 3 Unconstrained minimization
- 4 Inequality-constrained minimization
  - 4.1 Application to linear programming
- 5 Performance
- 6 Notes
- 7 Further reading
- 8 External links



An iteration of the ellipsoid method

## History

The ellipsoid method has a long history. As an [iterative method](#), a preliminary version was introduced by [Naum Z. Shor](#). In 1972, an [approximation algorithm](#) for real [convex minimization](#) was studied by [Arkadi Nemirovski](#) and David B. Yudin (Judin). As an algorithm for solving [linear programming](#) problems with rational data, the ellipsoid algorithm was studied by [Leonid Khachiyan](#): Khachiyan's achievement was to prove the [polynomial-time](#) solvability of linear programs.

Following Khachiyan's work, the ellipsoid method was the only algorithm for solving linear programs whose runtime had been proved to be polynomial until [Karmarkar's algorithm](#). However, the [interior-point method](#) and variants of the [simplex algorithm](#) are much faster than the ellipsoid method in practice. Karmarkar's algorithm is also faster in the worst case.

However, the ellipsoidal algorithm allows [complexity theorists](#) to achieve (worst-case) bounds that depend on the dimension of the problem and on the size of the data, but not on the number of rows, so it remained important in [combinatorial optimization](#) theory for many decades.<sup>[1][2][3][4]</sup> Only in the 21st century have interior-point algorithms with similar complexity properties appeared.<sup>[*citation needed*]</sup>

## Description

*Main article:* [Convex optimization](#)

A convex minimization problem consists of a [convex function](#)  $f_0(x) : \mathbb{R}^n \rightarrow \mathbb{R}$  to be minimized over the variable  $x$ , convex inequality constraints of the form  $f_i(x) \leq 0$ , where the functions  $f_i$  are convex, and linear equality constraints of the form  $h_i(x) = 0$ . We are also given an initial [ellipsoid](#)  $\mathcal{E}^{(0)} \subset \mathbb{R}^n$  defined as

$$\mathcal{E}^{(0)} = \left\{ z \in \mathbb{R}^n : (z - x_0)^T P_{(0)}^{-1} (z - x_0) \leq 1 \right\}$$

containing a minimizer  $x^*$ , where  $P \succ 0$  and  $x_0$  is the center of  $\mathcal{E}$ . Finally, we require the existence of a [cutting-plane](#) oracle for the function  $f$ . One example of a cutting-plane is given by a [subgradient](#)  $g$  of  $f$ .

## Unconstrained minimization

At the  $k$ -th iteration of the algorithm, we have a point  $x^{(k)}$  at the center of an ellipsoid

$$\mathcal{E}^{(k)} = \left\{ x \in \mathbb{R}^n : (x - x^{(k)})^T P_{(k)}^{-1} (x - x^{(k)}) \leq 1 \right\}.$$

We query the cutting-plane oracle to obtain a vector  $g^{(k+1)} \in \mathbb{R}^n$  such that  $g^{(k+1)T} (x^* - x^{(k)}) \leq 0$ . We therefore conclude that

$$x^* \in \mathcal{E}^{(k)} \cap \{ z : g^{(k+1)T} (z - x^{(k)}) \leq 0 \}.$$

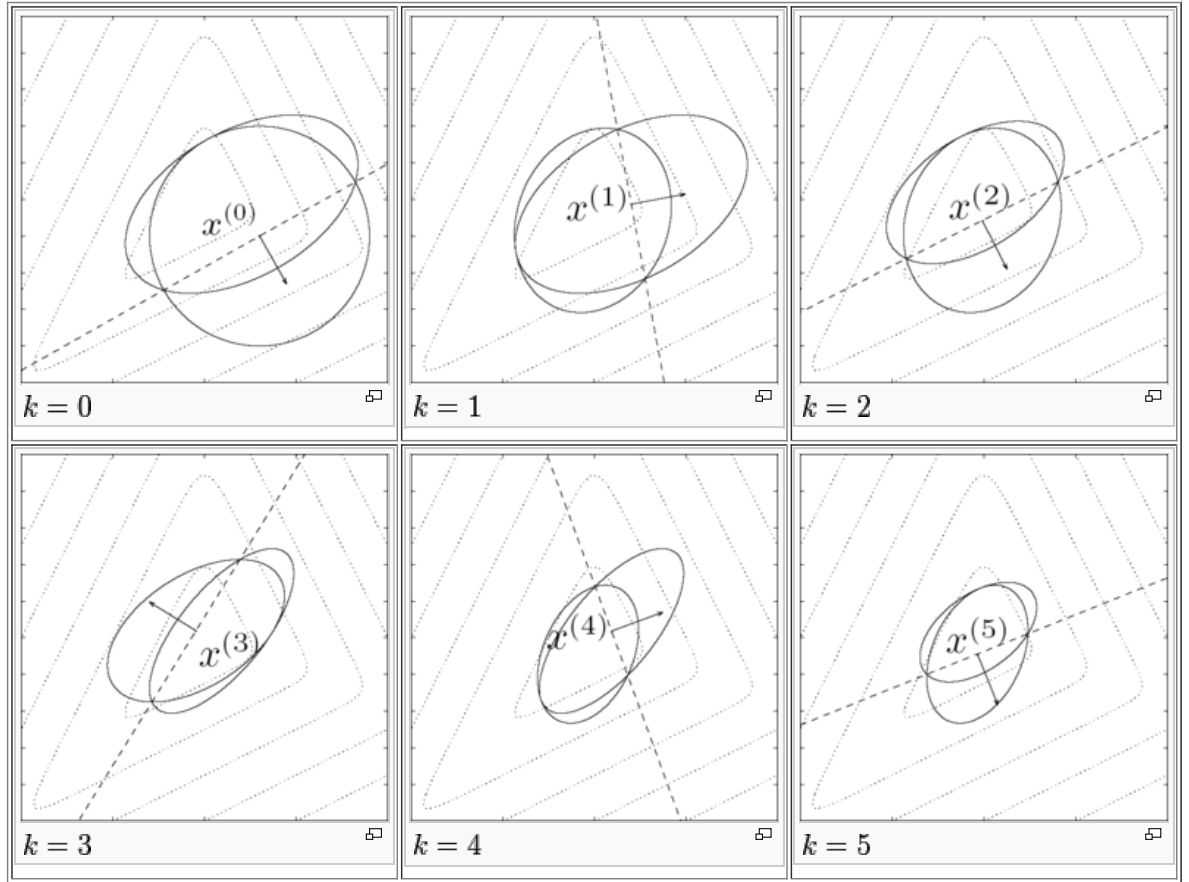
We set  $\mathcal{E}^{(k+1)}$  to be the ellipsoid of minimal volume containing the half-ellipsoid described above and compute  $x^{(k+1)}$ . The update is given by

$$x^{(k+1)} = x^{(k)} - \frac{1}{n+1} P_{(k)} \tilde{g}^{(k+1)}$$

$$P_{(k+1)} = \frac{n^2}{n^2-1} \left( P_{(k)} - \frac{2}{n+1} P_{(k)} \tilde{g}^{(k+1)} \tilde{g}^{(k+1)T} P_{(k)} \right)$$

where  $\tilde{g}^{(k+1)} = (1/\sqrt{g^{(k+1)T} P_{(k+1)} g^{(k+1)}}) g^{(k+1)}$ . The stopping criterion is given by the property that  $\sqrt{g^{(k)T} P_{(k)} g^{(k)}} \leq \epsilon \Rightarrow f(x^{(k)}) - f(x^*) \leq \epsilon$ .

Sample sequence of iterates



## Inequality-constrained minimization [\[edit\]](#)

At the  $k$ -th iteration of the algorithm for constrained minimization, we have a point  $x^{(k)}$  at the center of an ellipsoid  $\mathcal{E}^{(k)}$  as before. We also must maintain a list of values  $f_{\text{best}}^{(k)}$  recording the smallest objective value of feasible iterates so far.

Depending on whether or not the point  $x^{(k)}$  is feasible, we perform one of two tasks:

- If  $x^{(k)}$  is feasible, perform essentially the same update as in the unconstrained case, by choosing a subgradient  $g_0$  that satisfies

$$g_0^T (x^* - x^{(k)}) + f_0(x^{(k)}) - f_{\text{best}}^{(k)} \leq 0$$

- If  $x^{(k)}$  is infeasible and violates the  $j$ -th constraint, update the ellipsoid with a feasibility cut. Our feasibility cut may be a subgradient  $g_j$  of  $f_j$  which must satisfy

$$g_j^T (z - x^{(k)}) + f_j(x^{(k)}) \leq 0$$

for all feasible  $z$ .

## Application to linear programming [\[edit\]](#)

Inequality-constrained minimization of a function that is zero everywhere corresponds to the problem of simply identifying any feasible point. It turns out that any linear programming problem can be reduced to a linear feasibility problem (e.g. minimize the zero function subject to some linear inequality and equality constraints). One way to do this is by combining the primal and dual linear programs together into one program, and adding the additional (linear) constraint that the value of the primal solution is **no worse than** the value of the dual solution. Another way is to treat the objective of the linear program as an additional constraint, and use binary search to find the optimum value.<sup>[\[citation needed\]](#)</sup>

## Performance [\[edit\]](#)

The ellipsoid method is used on low-dimensional problems, such as planar location problems, where it is **numerically stable**. On even "small"-sized problems, it suffers from numerical instability and poor performance in practice.

However, the ellipsoid method is an important theoretical technique in [combinatorial optimization](#). In [computational complexity theory](#), the ellipsoid algorithm is attractive because its complexity depends on the number of columns and the digital size of the coefficients, but not on the number of rows. In the 21st century, interior-point algorithms with similar properties have appeared<sup>[*citation needed*]</sup>.

Notes [edit]

- 1. ↑ M. Grötschel, [L. Lovász](#), [A. Schrijver](#): *Geometric Algorithms and Combinatorial Optimization*, Springer, 1988.
- 2. ↑ [L. Lovász](#): *An Algorithmic Theory of Numbers, Graphs, and Convexity*, CBMS-NSF Regional Conference Series in Applied Mathematics 50, SIAM, Philadelphia, Pennsylvania, 1986.
- 3. ↑ V. Chandru and M.R.Rao, Linear Programming, Chapter 31 in *Algorithms and Theory of Computation Handbook*, edited by [M. J. Atallah](#), CRC Press 1999, 31-1 to 31-37.
- 4. ↑ V. Chandru and M.R.Rao, Integer Programming, Chapter 32 in *Algorithms and Theory of Computation Handbook*, edited by M.J.Atallah, CRC Press 1999, 32-1 to 32-45.

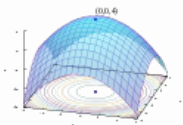
Further reading [edit]

- Dmitris Alevras and Manfred W. Padberg, *Linear Optimization and Extensions: Problems and Extensions*, Universitext, Springer-Verlag, 2001. (Problems from Padberg with solutions.)
- V. Chandru and M.R.Rao, Linear Programming, Chapter 31 in *Algorithms and Theory of Computation Handbook*, edited by M.J.Atallah, CRC Press 1999, 31-1 to 31-37.
- V. Chandru and M.R.Rao, Integer Programming, Chapter 32 in *Algorithms and Theory of Computation Handbook*, edited by M.J.Atallah, CRC Press 1999, 32-1 to 32-45.
- [George B. Dantzig](#) and Mukund N. Thapa. 1997. *Linear programming 1: Introduction*. Springer-Verlag.
- [George B. Dantzig](#) and Mukund N. Thapa. 2003. *Linear Programming 2: Theory and Extensions*. Springer-Verlag.
- M. Grötschel, [L. Lovász](#), [A. Schrijver](#): *Geometric Algorithms and Combinatorial Optimization*, Springer, 1988
- [L. Lovász](#): *An Algorithmic Theory of Numbers, Graphs, and Convexity*, CBMS-NSF Regional Conference Series in Applied Mathematics 50, SIAM, Philadelphia, Pennsylvania, 1986
- Kattta G. Murty, *Linear Programming*, Wiley, 1983.
- M. Padberg, *Linear Optimization and Extensions*, Second Edition, Springer-Verlag, 1999.
- [Christos H. Papadimitriou](#) and Kenneth Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Corrected republication with a new preface, Dover.
- [Alexander Schrijver](#), *Theory of Linear and Integer Programming*. John Wiley & sons, 1998, ISBN 0-471-98232-6

External links [edit]

- [EE364b](#) , a Stanford course homepage

v · t · e		<b>Optimization: Algorithms, methods, and heuristics</b>	<span>[hide]</span>
		<b>Unconstrained nonlinear: Methods calling ...</b>	<span>[show]</span>
		<b>Constrained nonlinear</b>	<span>[show]</span>
		<b>Convex optimization</b>	<span>[hide]</span>
<b>Convex minimization</b>	Cutting-plane method · Reduced gradient (Frank–Wolfe) · Subgradient method		
<b>Linear and quadratic</b>	<b>Interior point</b>	<b>Ellipsoid algorithm of Khachiyan</b> · Projective algorithm of Karmarkar	
	<b>Basis-Exchange</b>	Simplex algorithm of Dantzig · Revised simplex algorithm · Criss-cross algorithm · Principal pivoting algorithm of Lemke	
		<b>Combinatorial</b>	<span>[show]</span>
		<b>Metaheuristics</b>	<span>[show]</span>
		<b>Categories</b> (Algorithms and methods · Heuristics) · <b>Software</b>	



Categories: [Combinatorial optimization](#) | [Operations research](#) | [Convex optimization](#) | [Linear programming](#)