Computational Issues in the Numerical Solution of Optimization Problems Defined by Expensive Computer Simulations[1]

Virginia Torczon
Department of Computer Science
College of William & Mary

[1]With thanks to the National Science Foundation for support under Grant CCR-9734044

Collaborators:

Robert Michael Lewis
Institute for Computer Applications
in Science and Engineering (ICASE)
NASA Langley Research Center

Michael W. Trosset
Department of Mathematics
College of William & Mary

Problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathbf{R}^n, \end{array}$$

where $f : \mathbf{R}^n \to \mathbf{R}$.

This is the *unconstrained* problem.

There is a suite of results (with Robert Michael Lewis) to handle classes of constrained problems of the general form:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in S \subseteq \mathbf{R}^n. \end{array}$$

Operating Assumptions:

- Evaluation of the objective f depends on the output of computer simulations that may involve large numbers of state or system variables.

  The objective f is *not* a simple algebraic expression exclusively in terms of the variables x.

- The objective may be expensive to compute because the simulations upon which its value is based are expensive to compute.

The *pattern search* algorithms we will discuss are applicable to general nonlinear optimization problems, but the motivation for their study are problems with the above features.

Consequences:

- Simulations of complex physical processes can lead to:
  - nonstochastic errors in the evaluation of the objective due to the approximation, rounding, and truncation introduced by the simulations and
  - high-frequency, low-amplitude distortions in the "true" value of the objective,

  features which can cause difficulties for higher-order optimization methods.

- If the simulation is expensive to compute, we also must be frugal in our evaluation of the objective.

Difficulties:

- Software for computing derivatives of the objective may not be available.
- Errors in the evaluation of the objective make it difficult—often impossible—to obtain reliable finite-difference estimates of the directional derivative.
- The high-frequency, low-amplitude distortions often lead higher-order methods to spurious local minimizers.
- For some values of x, the simulation may fail—even when x satisfies the explicitly stated constraints.
- Often there are multiple local minimizers and the user wants a global—or at least "more" global—solution.

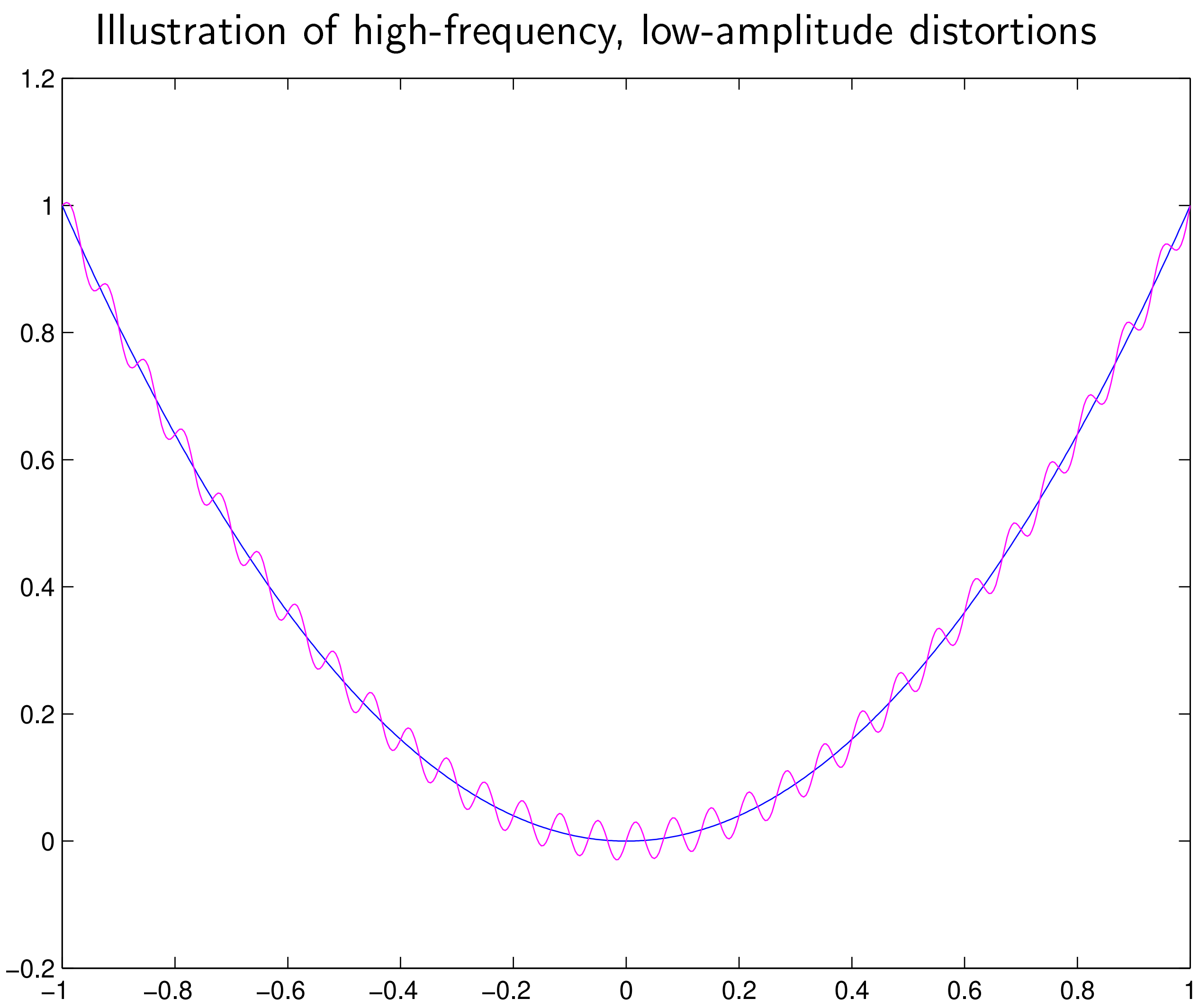


Illustration of high-frequency, low-amplitude distortions
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1

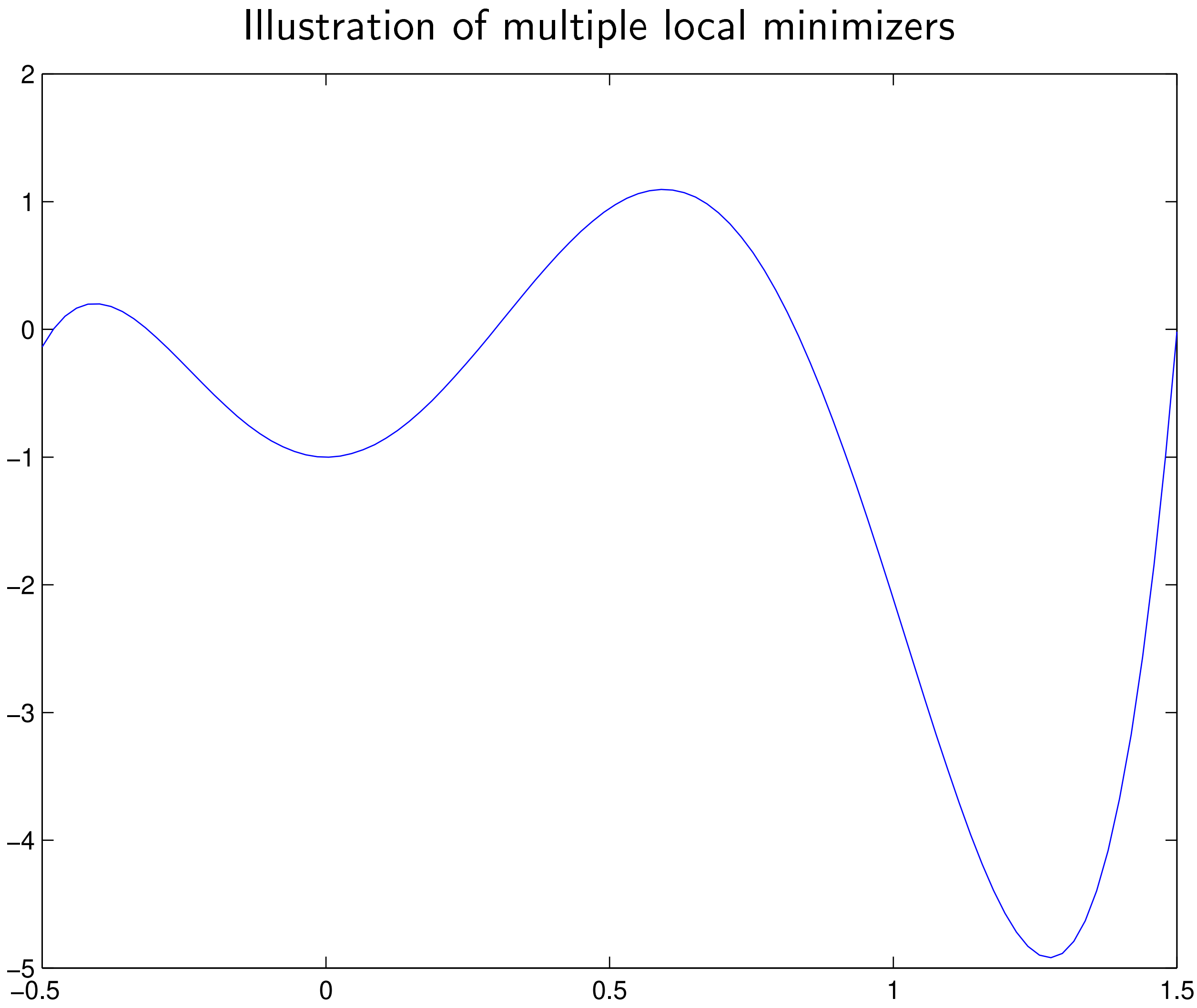


Illustration of multiple local minimizers
2
1
0
−1
−2
−3
−4
−5
−0.5
0
0.5
1
1.5

Classical Approach:

Use a modified form of Newton's Method (a so-called quasi-Newton method).

Solve

$$Bd_k^N = -\nabla f(x_k)$$

for the Newton *direction* d_k^N, where

- B is an approximation to $\nabla^2 f(x_k)$, the Hessian of f at the current iterate x_k, and

- $\nabla f(x_k)$ is the directional derivative (or an approximation) of f at x_k.

Form the *step* $s_k = \Delta_k d_k^N$, $\Delta_k \in (0, 1]$.

If s_k is *acceptable*, form a new *iterate* $x_{k+1} = x_k + s_k$.

Repeat until

$$\|\nabla f(x_k)\| \leq \epsilon,$$

where ϵ is a stopping tolerance chosen by the user.

Practical Concern:

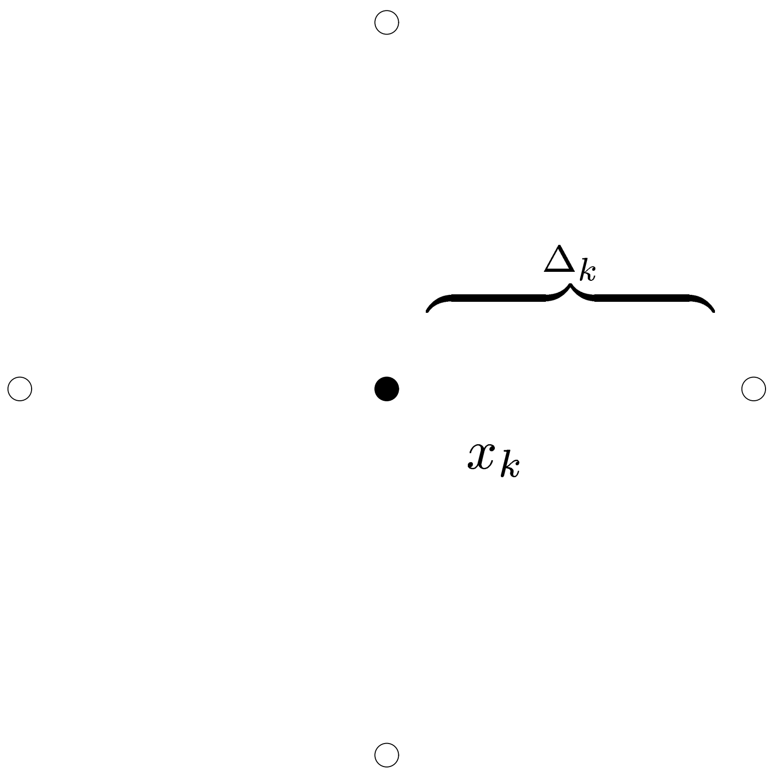
For quasi-Newton methods, the directional derivative $\nabla f(x_k)$ plays a pivotal role:

- In forming the system to determine the direction d_k^N.
- Typically, in forming the approximation B of the Hessian $\nabla^2 f(x_k)$.
- In judging the *acceptability* of the step s_k.
- In testing for termination.

What do we do if we do not have—and cannot reliably estimate—$\nabla f(x_k)$*?*

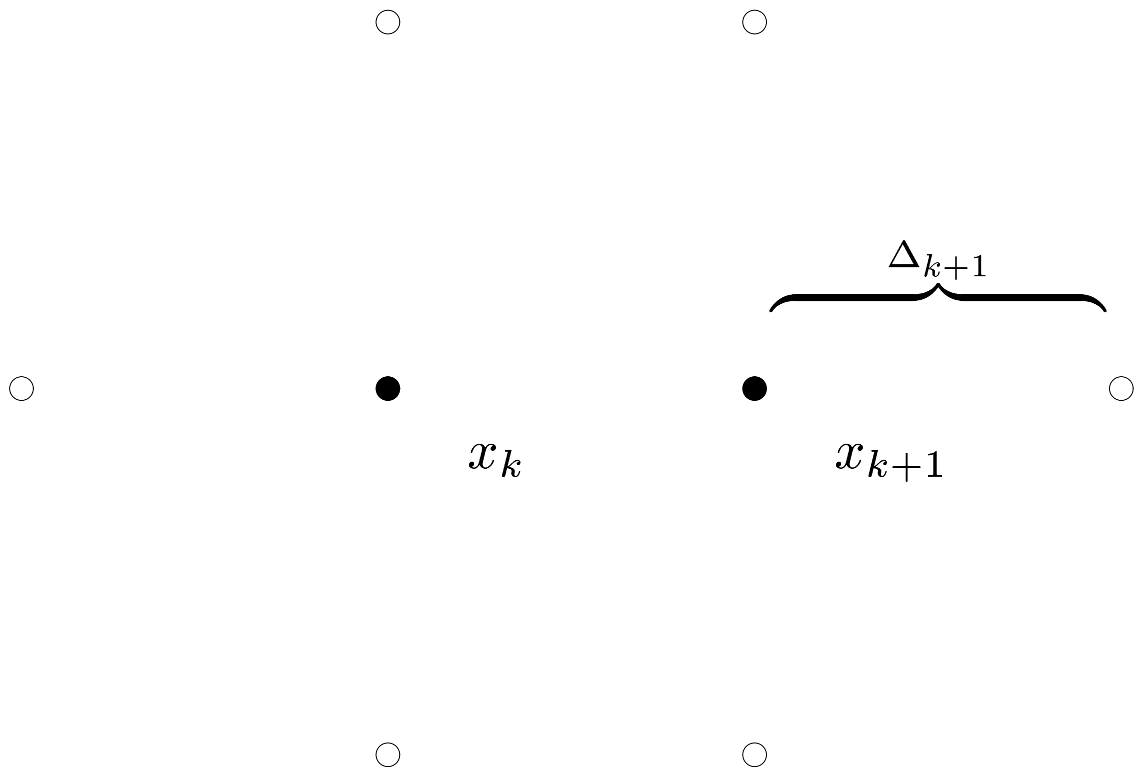
Pattern Search

The simplest, provably convergent, optimization algorithm:



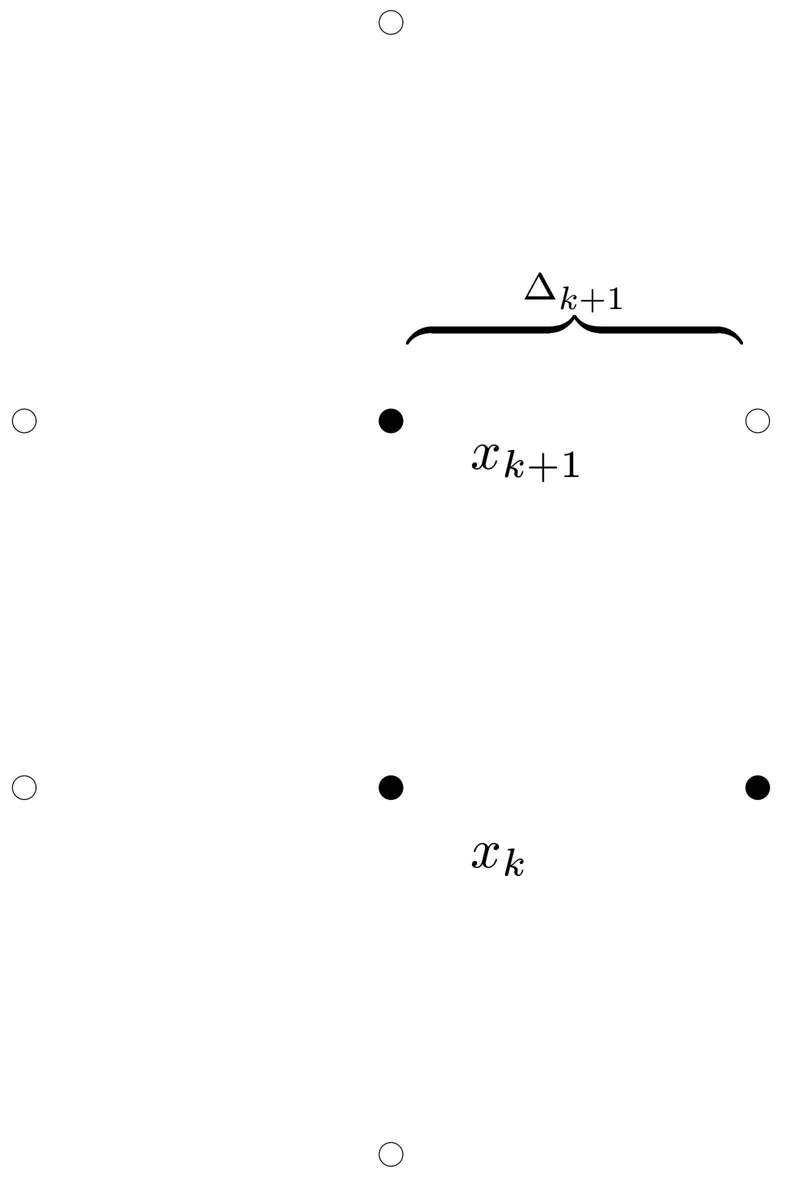
Compass Search

First possible outcome:



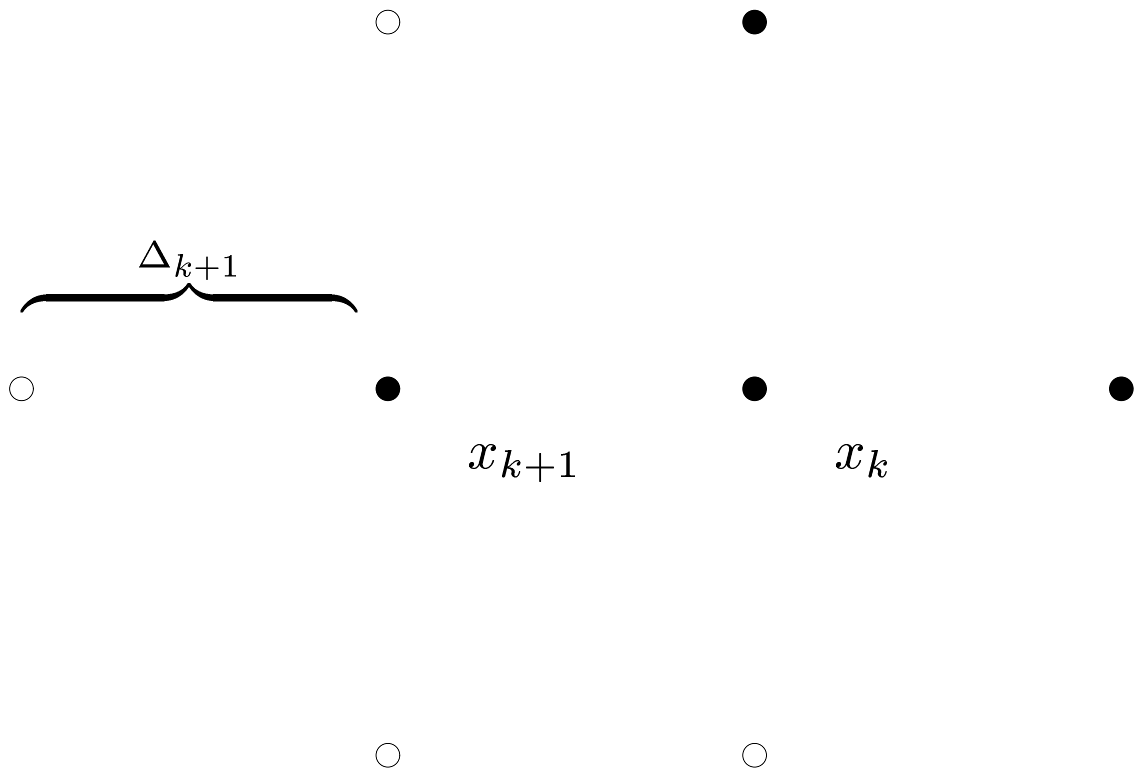
Compass Search

Second possible outcome:



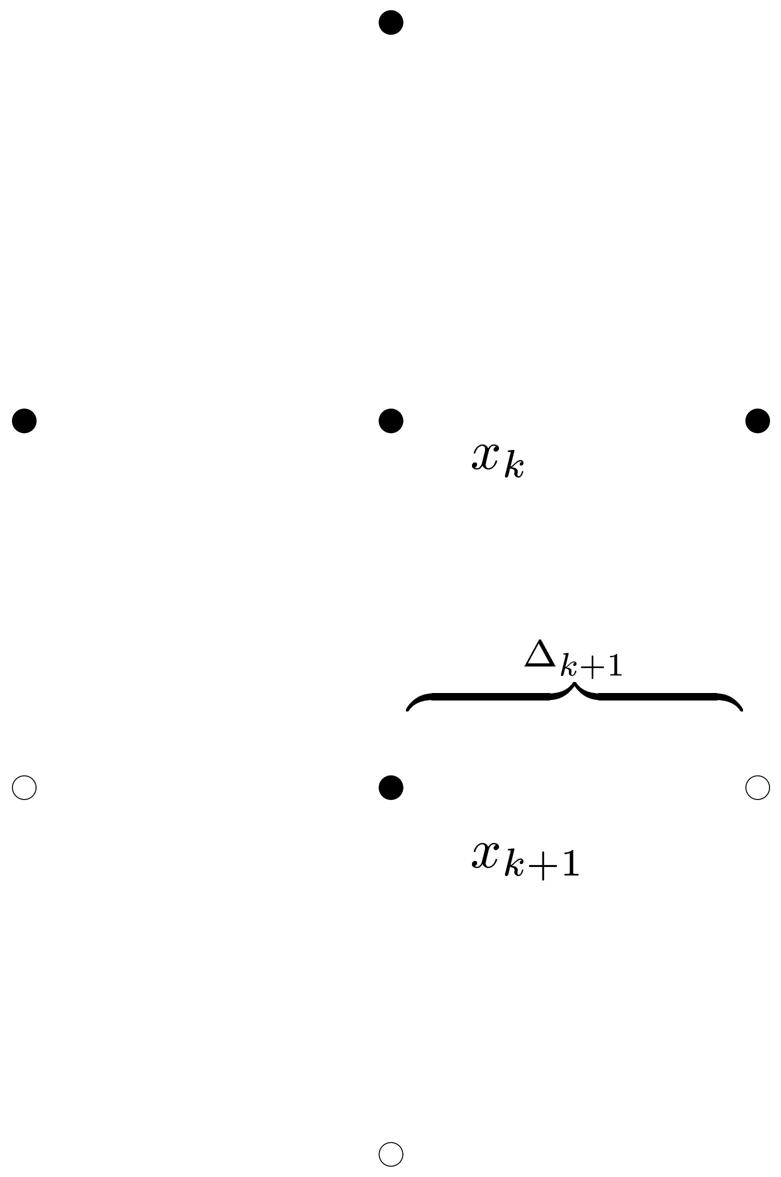
Compass Search

Third possible outcome:



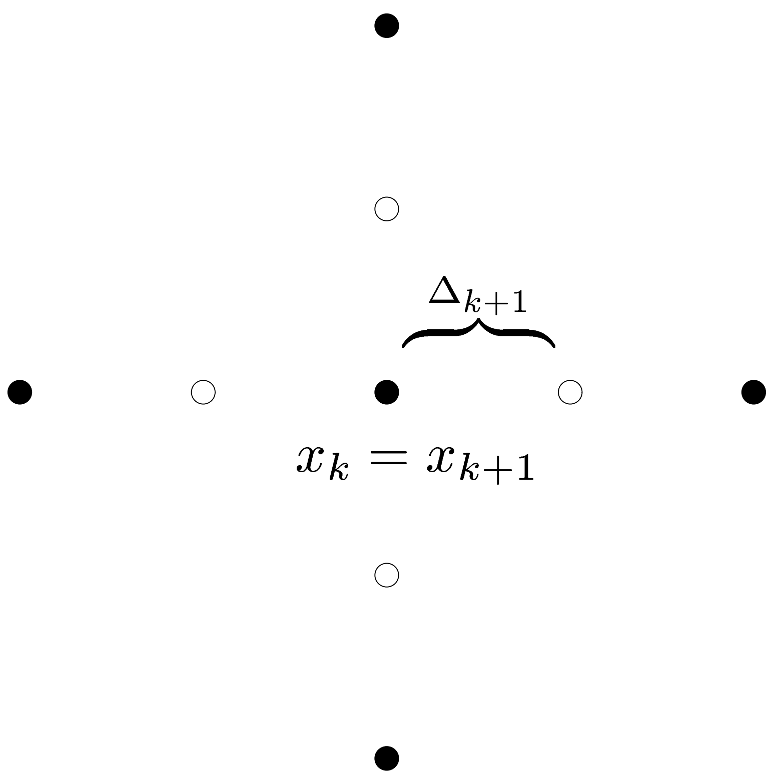
Compass Search

Fourth possible outcome:



Compass Search

The fallback strategy:



Compass Search

Generalized Pattern Search:

Given $x_0 \in \mathbf{R}^n$, $f(x_0)$, $D_0 \in \mathbf{R}^{n \times p_0}$, $\Delta_0 > 0$, and ϵ, for $k = 0, 1, \ldots$ until $\Delta_k < \epsilon$ do $\{$

1. Find a step $s_k = \Delta_k d_k$ using the procedure `Exploratory_Moves`(Δ_k, D_k).

2. If $f(x_k + s_k) < f(x_k)$ then $x_{k+1} = x_k + s_k$; otherwise, $x_{k+1} = x_k$.

3. `Update`(Δ_k, D_k).

$\}$

Practical Considerations:

For a pattern search method:

- The direction d_k is chosen from a predefined set of directions D_k.

- The only condition for accepting a step $s_k = \Delta_k d_k$ from the set of possible candidates is that

$$f(x_k + s_k) < f(x_k)$$

  (i.e., we require *simple* decrease on the value of f at x_k).

- We terminate this iterative process when

$$\Delta_k \leq \epsilon,$$

  where ϵ is a stopping tolerance chosen by the user.

The directional derivative does not play an explicit role in the process.

Global Convergence Results:

For both (globalized) quasi-Newton methods and pattern search methods, either

$$\liminf_{k\to\infty} \|\nabla f(x_k)\| \quad = \quad 0$$

or

$$\lim_{k\to\infty} \|\nabla f(x_k)\| \quad = \quad 0$$

(depending on additional assumptions we place on either the problem or the implementation).

How can this be?!?

Ingredients for Analysis:

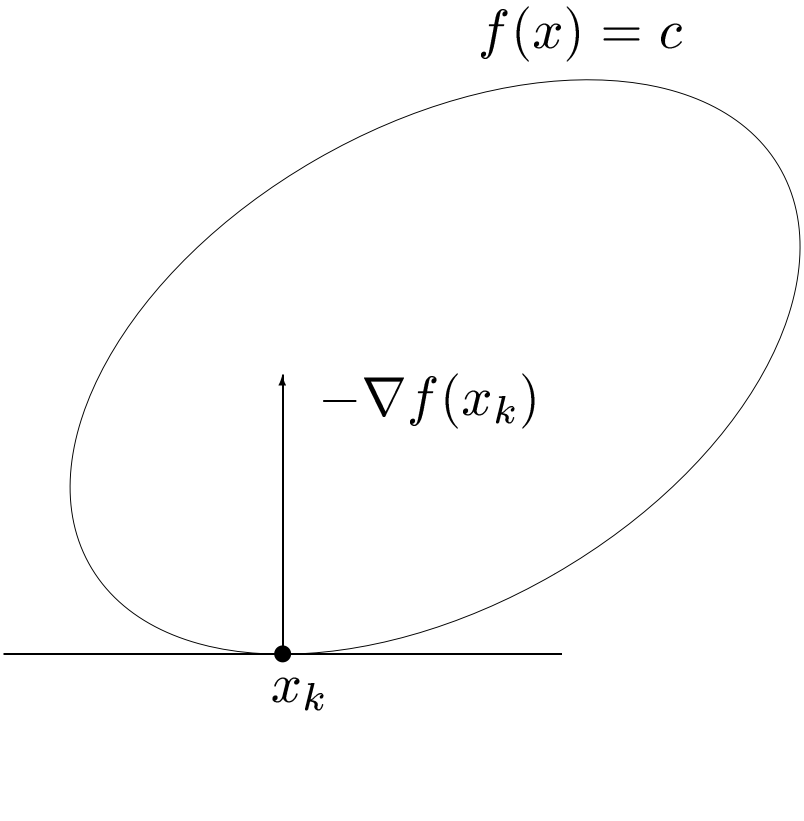
To prove global convergence for a quasi-Newton method using a line search globalization strategy, we need to show the following:

1. We have a direction of descent.

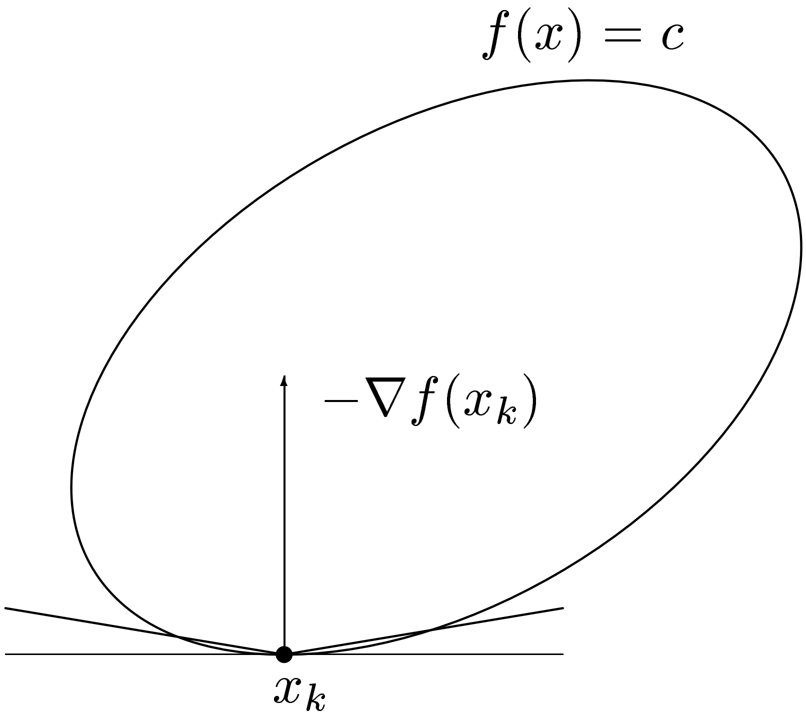
2. The direction of descent is is uniformly bounded away from orthogonality with the gradient.

3. The step produces *sufficient* decrease on f at x_k.

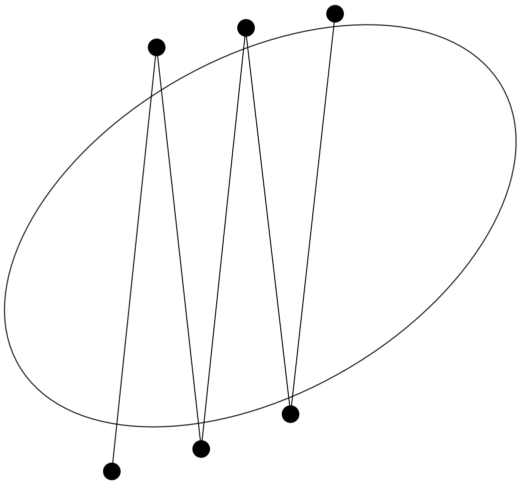
1. A Direction of Descent:



2. Bounding the Search Direction:

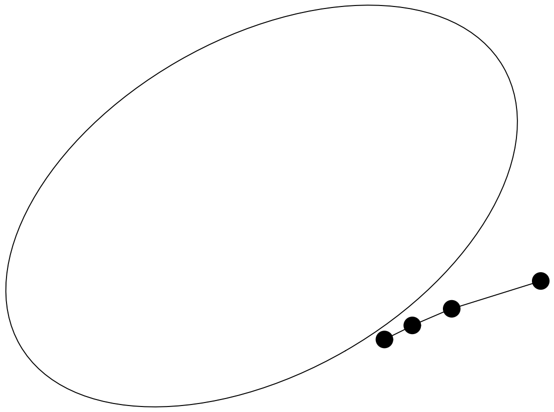


Possible Pathologies:



decrease too small
relative to the
length of the step

$\Longrightarrow$ need for step length control



decrease too small
relative to the
norm of the gradient

3. Classical Step Length Control:

The Armijo–Goldstein–Wolfe conditions for line search globalization strategies:

Given a suitable descent direction d_k, choose a step length $\Delta_k > 0$ such that for some fixed $\alpha \in (0, 1)$ and fixed $\beta \in (\alpha, 1)$,

$$x_{k+1} = x_k + \Delta_k d_k$$

satisfies both

$$f(x_{k+1}) \leq f(x_k) + \alpha \Delta_k \nabla f(x_k)^T d_k$$

and

$$\nabla f(x_{k+1})^T d_k \geq \beta \nabla f(x_k)^T d_k.$$

These are referred to as *sufficient* decrease conditions on the acceptability of the step $s_k = \Delta_k d_k$.

Summary:

The devices that ensure the global convergence of line search methods for unconstrained minimization are:

1. The choice of suitably good descent direction.

2. Step-length control:

   (a) a mechanism to avoid steps that are too long, and
   (b) a mechanism to avoid steps that are too short.

These devices make explicit use of $\nabla f(x_k)$ and are built into all modern implementations of quasi-Newton methods with a line search globalization strategy.

Why Pattern Search Works:

1. At each iteration, we require D_k to include a *sufficient* number of directions to ensure a direction of descent.

2. We require the set of possibilities from which we choose D_k to be finite to ensure the existence of at least one direction that is uniformly bounded away from orthogonality with the gradient.

3. We implement a reasonable back-tracking strategy that avoids unnecessarily short steps. Otherwise we avoid unsuitable steps by restricting the nature of the steps allowed.

1. Employ a Positive Basis

For pattern search, we require that the set of directions D_k contain a positive basis Π_k for $\mathbf{R}^n$.

The *positive span* of a set of vectors $\{a_1, \ldots, a_r\}$ is the cone:

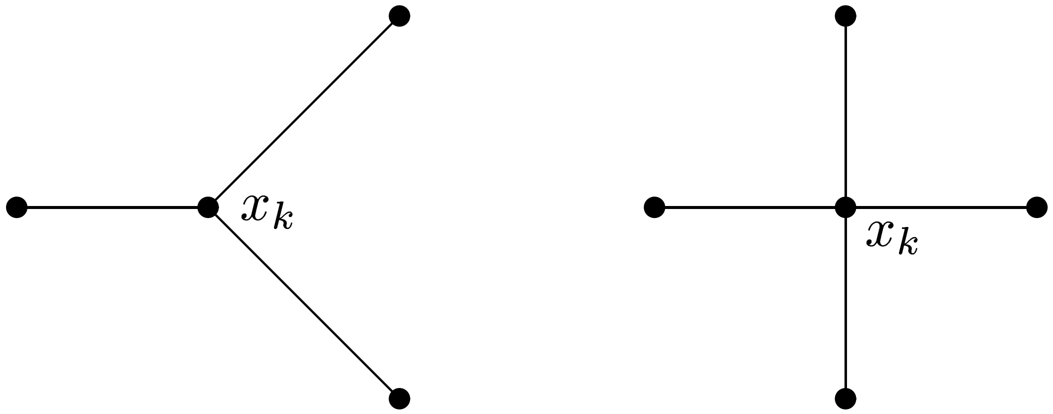
$$\{a \in \mathbf{R}^n | a = c_1 a_1 + \cdots + c_r a_r,\ c_i \geq 0 \text{ for all } i\}$$

The set $\{a_1, \ldots, a_r\}$ is called *positively dependent* if one of the a_i's is a nonnegative combination of the others; otherwise the set is *positively independent*.

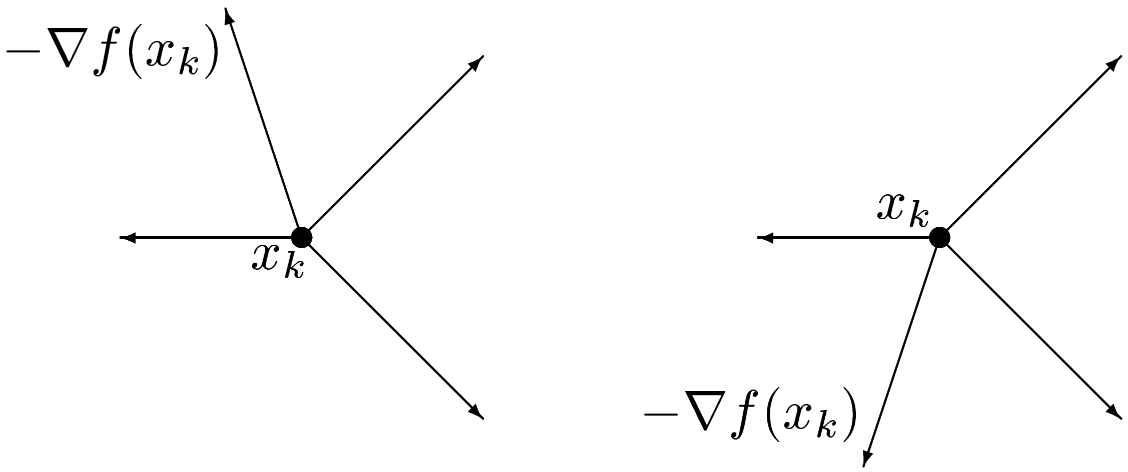
A *positive basis* is a positively independent set whose positive span is $\mathbf{R}^n$.

A positive basis must contain at least $n + 1$ vectors and can contain no more than $2n$ vectors; we refer to the former as *minimal* and the latter as *maximal*.

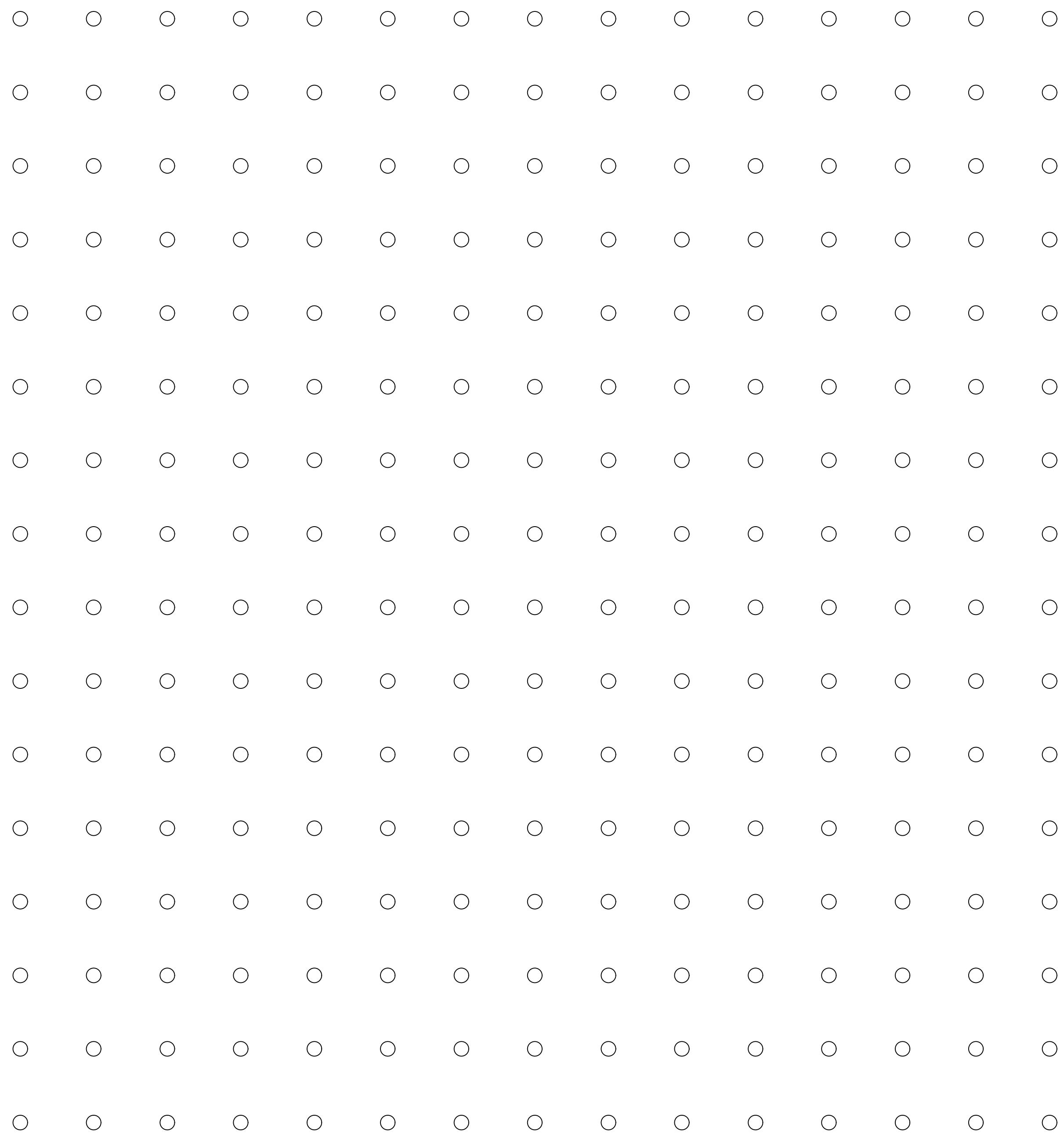
A Minimal and a Maximal Positive Basis



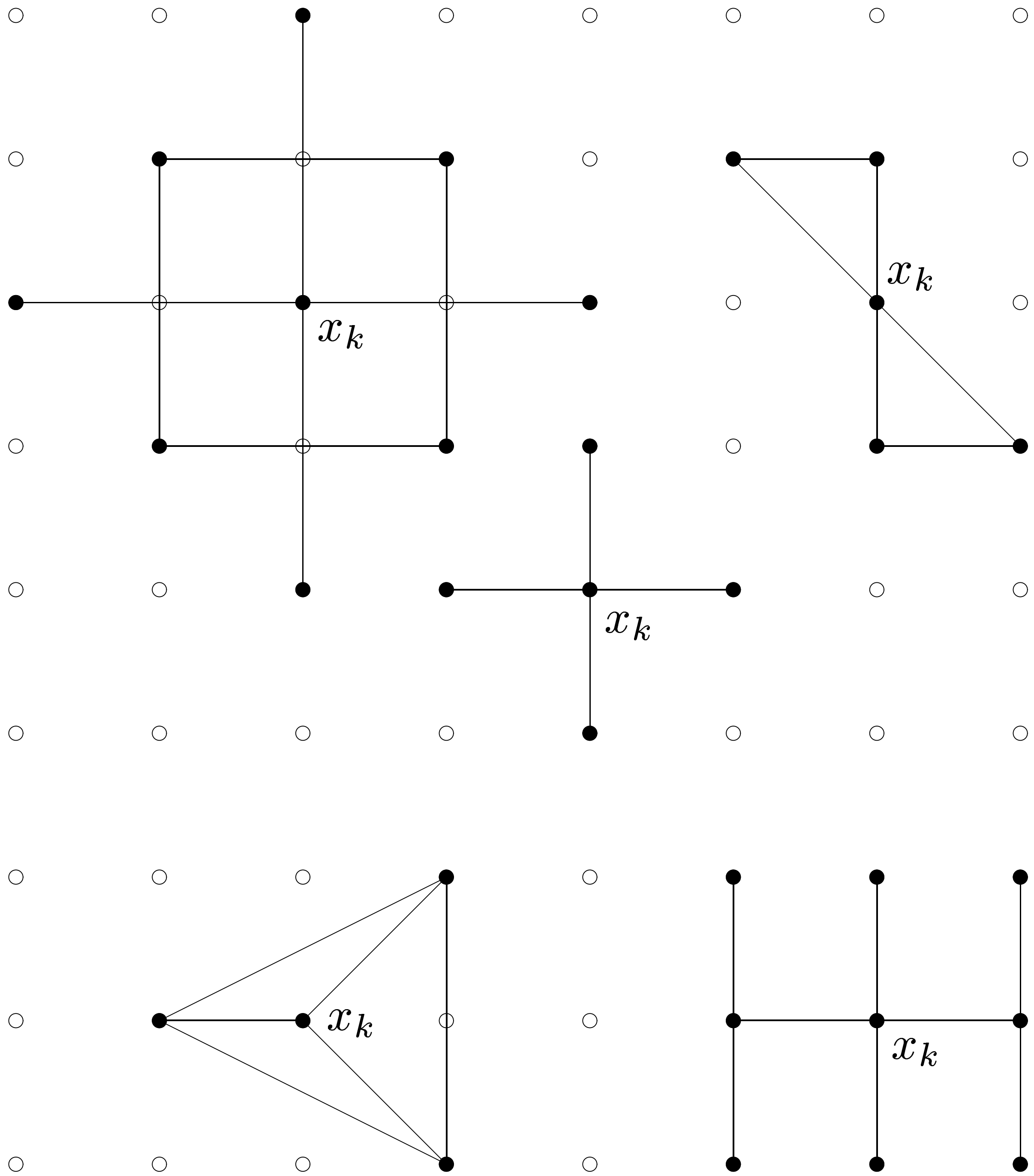
2. Bounding the Directions:



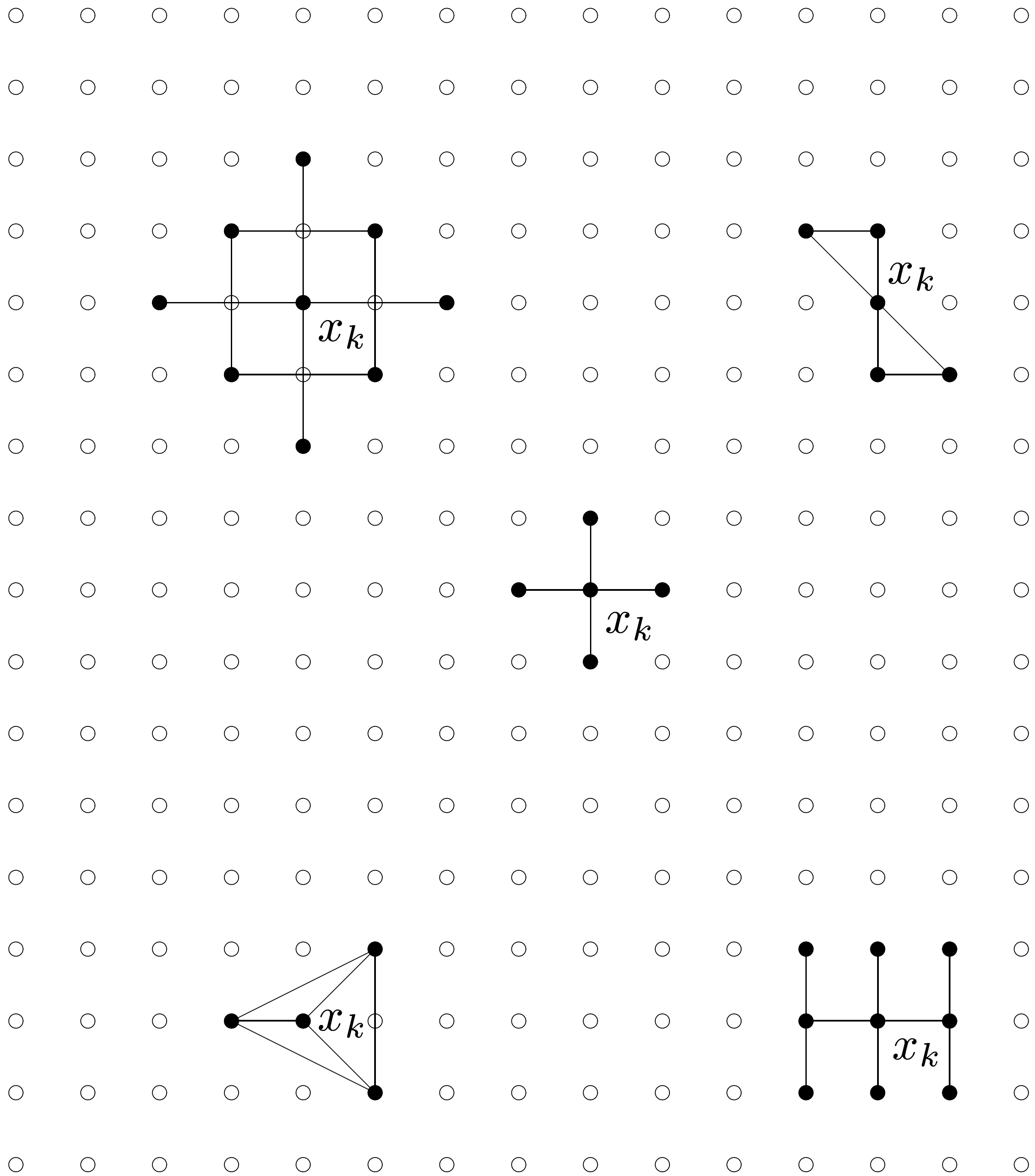
3. The Underlying Lattice



Some possible patterns:



On a refinement of the lattice:



The `Exploratory Moves()` procedure

The `Exploratory Moves()` procedure must satisfy two simple conditions:

1. The step returned must be an element of $\Delta_k D_k$.

2. The step s_k must satisfy either

$$f(x_k + s_k) < f(x_k)$$

   or

$$s_k = 0.$$

Furthermore s_k may be 0 only if none of the step in $\Delta_k \Pi_k$ yielded decrease on $f(x_k)$.

The first condition prevents arbitrary steps along arbitrary directions.

The second condition is a back-tracking control mechanism that prevents us from taking shorter steps unless it is truly necessary to do so to realize descent.

Why the Theory Matters

- There are clearly strong practical reasons to be interested in methods that do not require us to know or estimate $\nabla f(x_k)$.

- Once we understand the conditions necessary to ensure convergence of pattern search methods, we are free to design *new* variants that allow us to exploit the features of the computational environment that is available to us, the features the problem that is of interest to us—or both.

- We can better explain the numerical behavior of these algorithms.

Examples:

- Direct search methods on parallel machines
  J. E. Dennis, Jr. and Virginia Torczon
  SIAM Journal on Optimization,
  Volume 1, Number 4, pages 448–474, 1991.

- Rank ordering and positive bases
  in pattern search algorithms
  Robert Michael Lewis and Virginia Torczon
  In revision for *Mathematical Programming*.

- A stationary point convergence theory for evolutionary algorithms
  William E. Hart
  Foundations of Genetic Algorithms 4, 1996.

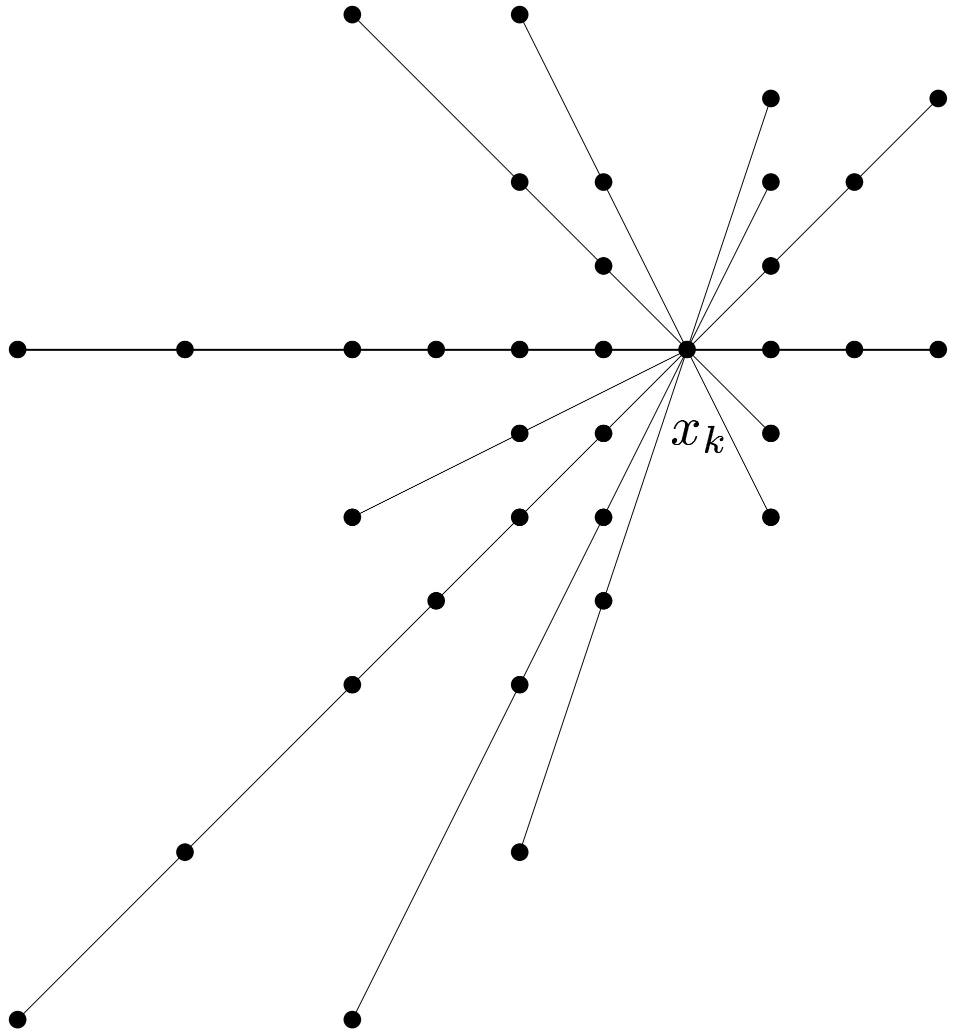
- Numerical optimization using computer experiments
  Michael W. Trosset and Virginia Torczon
  ICASE Technical Report 97–38.

- A rigorous framework for the optimization of expensive functions by surrogates
  Andrew J. Booker, J. E. Dennis, Jr., Paul D. Frank, David B. Serafini, Virginia Torczon, and Michael W. Trosset
  Structural Optimization
  Volume 17, Number 1, pages 1–13, 1999.

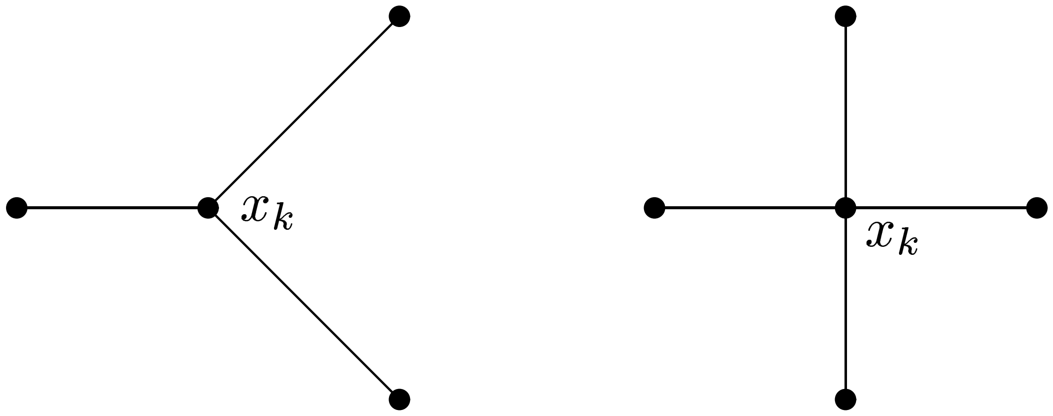
- Asynchronous parallel direct search methods for nonlinear optimization
  Patricia D. Hough, Tamara G. Kolda, and Virginia Torczon
  In preparation.

- The numerical behavior of variants of pattern search
  Elizabeth D. Dolan
  Senior Honors Thesis
  Department of Computer Science
  College of William & Mary, 1999.

More aggressive patterns



Minimal versus maximal patterns



What to expect from pattern search methods:

- useful exploratory tool
  (wide applicability, ease of use)

- good global behavior
  (they are gradient-related methods)

- slow local convergence
  (no second-order information)

- potential for large number of function evaluations
  (they are sampling methods)

- effective for problems with a relatively small number of variables
  (curse of dimensionality)

When to use pattern search methods:

- evaluation of f is inaccurate
- derivatives of f are either not available or not reliable
- f is highly nonlinear
- only rank (order) information is available
- want a "wider" search of the design space
- f is nondifferentiable
- the first partial derivatives of f are discontinuous
- access to parallel or distributed computing platform

Problems solved:

- chemical process control
- seismic inversion
- hearing recovery after exposure to loud noise
- control of robotic arm
- airfoil design
- vibration suppression
- helicopter rotor blade design
-
-
-

Web Resources:

For PostScript files of related papers, the PostScript file containing this talk, or a PostScript file for the article upon which much of this talk is based (*Why Pattern Search Works*), visit:

`http://www.cs.wm.edu/~va`

and click on one of:

- `Publications`,
- `presentations`, or
- `articles`.