Non-Derivative Optimization: Mathematics or Heuristics?

Margaret H. Wright

Computer Science Department

Courant Institute of Mathematical Sciences

New York University

Kovalevskaya Colloquium

Berlin Mathematical School

June 19, 2009

Thank you for the great honor of inviting me to give the Kovalevskaya Colloquium.

Sofia Kovalevskaya
1850–1891

Kovalevskaya's connections with my title?

Mathematics (close): In her own words, she had

> . . . a reverence for mathematics as an *exalted and mysterious* science which opens up to its initiates a new world of wonder, *inaccessible to ordinary mortals*. . . . [Emphasis added.]

Heuristics (not close!): The *Oxford English Dictionary* dates the first use of "heuristic" to 1860 (during her lifetime). According to Pólya (1957):

> Heuristic reasoning is reasoning not regarded as final and strict but as provisional and plausible only, whose purpose is to *discover the solution to the present problem*. [Emphasis added.]

Today's general topic: solving the unconstrained continuous optimization problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x)$$

And not just any old f.

Many real-world applications involve extremely difficult ("nasty") optimization problems in which f has one or more of the following properties:

- f is **very** time-consuming or expensive to calculate, even on the highest-end machines, or it may involve data collected from the real world (which may take hours, days, weeks, . . .)
- f is unpredictably non-nice (e.g., undefined at certain points, discontinuous, non-smooth)
- f is evaluated by "black-box" software whose inner workings are not under the current user's control

- f is “noisy” because of
  - adaptivity in calculations
  - stochastic elements
  - uncontrollable variations (e.g., inclusion of real-world experimental data)

In cases like these, first derivatives are difficult or impossible to obtain, even with advanced automatic differentiation.

A few examples (among hundreds):

- Drug selection during cancer chemotherapy, based on the patient's measured responses;
- Design of wireless systems;
- Design of heating, air-conditioning, and ventilation (HVAC) systems;
- Modeling the population dynamics of the cannibalistic flour beetle;
- Estimating the structure of transmembrane proteins;
- Circuit design.

Unless we give up immediately, the only choice is a **non-derivative** method, meaning a method that:

1. Uses only function values (no derivatives);
2. Does not, in its heart, approximate the gradient.

But what does "approximate the gradient" really mean?

Two broad classes of non-derivative methods:

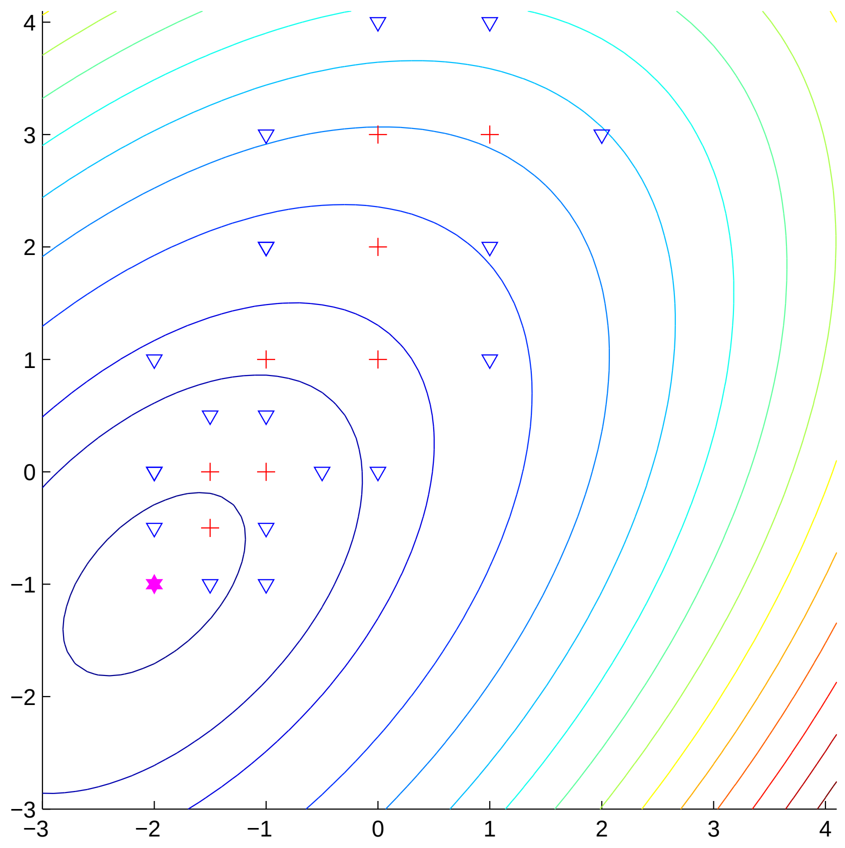
1. **Model-based**
   - Create a model of f, usually quadratic, based on interpolation or least-squares, and minimize the model (à la Newton's method or quasi-Newton methods)
   - Some smoothness assumed somewhere.
2. $**$ **Geometry-based** $**$
   - No *explicit* model of f, but sometimes used with "surrogate" models

NB: Genetic and evolutionary algorithms are not considered here.

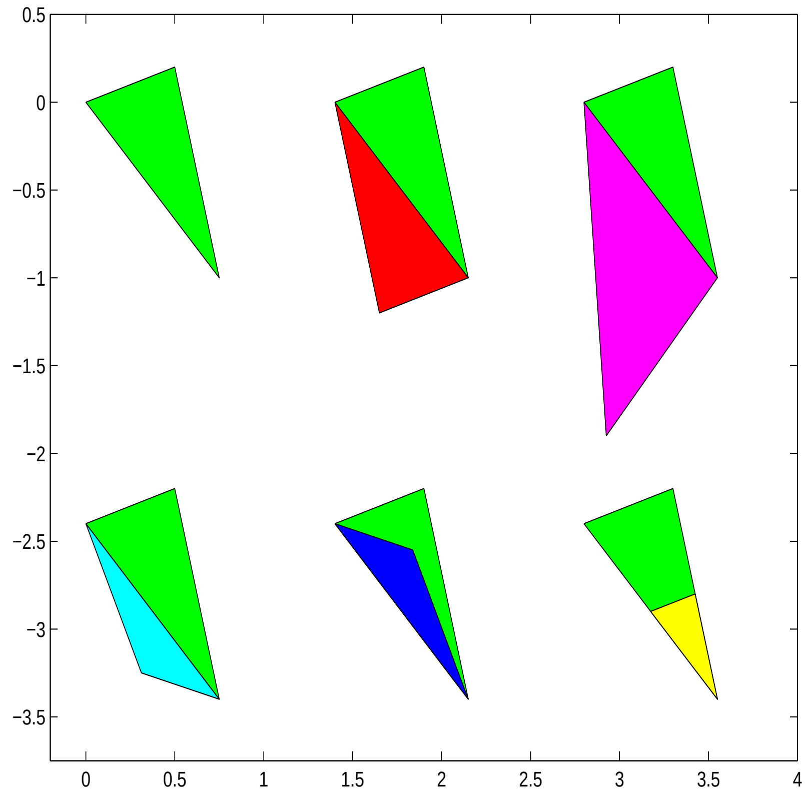
A sketchy history of non-derivative optimization methods:

- Started in 1950s (or before)—Fermi and Metropolis applied coordinate search in a 1952 paper.
- LOVED by practitioners from day 1, especially the "simplex" method of Nelder and Mead (1965), of which more later.
- Often based on **low-dimensional** intuition.

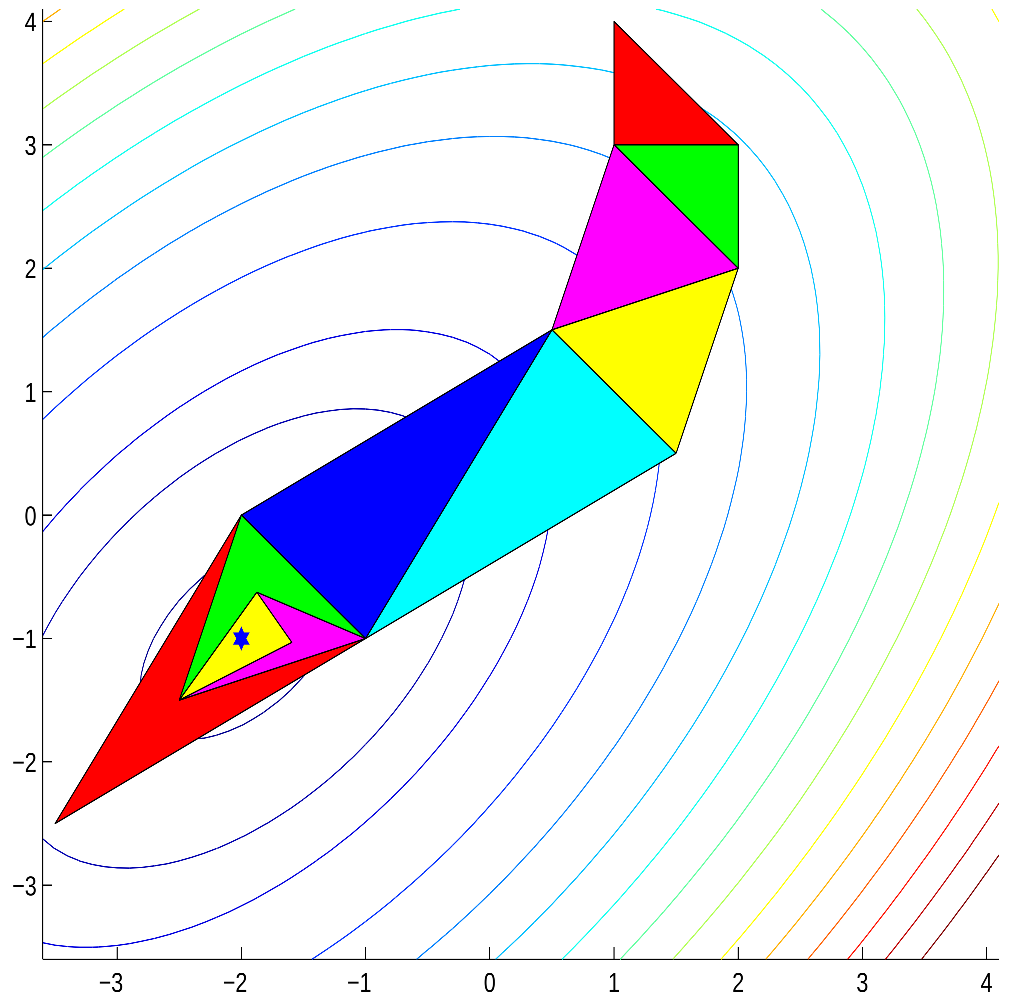
"Opportunistic" coordinate search (Fermi and Metropolis); keep looking for a *new best point* by searching along the $\pm$ coordinate directions, shrinking the step when no strictly better point is found.



The most popular non-derivative method: the Nelder–Mead "simplex" method (1965), in which each iteration is associated with a nondegenerate simplex in $\mathcal{R}^n$.



Goal of each iteration: Find a new vertex where f is strictly better than the *worst* vertex.


4
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
4

Starting in the mid-1960s, non-derivative methods became essentially invisible within the *mainstream optimization community*, for two reasons:

1. No theory!

Hence, because of the "mathematization" of optimization, by the mid-1970s non-derivative methods were not just ignored, but were actually scorned by mainstream optimizers.

2. (Perceived?) difficulties in practice.

Geometry-based methods were observed to be very slow at times, with performance tending to degrade as the problem dimension increases. Model-based methods were regarded as much less efficient than finite-difference gradient-based methods.

Nonetheless, despite a lack of theory and occasional poor behavior, through the 1980s and into the 1990s, non-derivative methods (almost always Nelder-Mead) were the most popular optimization routines in software libraries, i.e. users loved non-derivative methods.

Why would users be so foolish as to ignore the experts?

One plausible explanation for their popularity: users did not want to write code to evaluate derivatives.

Don't be too quick to jeer at this reason!

$$f(x) = \frac{\sinh(2x_1^2 \cos x_2) \arctan x_3}{\sqrt{x_3^4 x_5^2 + 1} - x_4}$$

In those days, the biggest single error in using optimization software was programming the derivatives incorrectly.

This should not be a big issue today because we have reliable and sophisticated software for automatic differentiation.

A second major appeal of geometry-based non-derivative methods, as we've already seen:

Some of them are simple to describe in 2-d pictures so that they appeal to (low-dimensional) intuition.

So these methods seem to be "accessible to ordinary mortals", a property that finds favor with algorithm/software users even though it is contrary to Kovalevskaya's description of the world of mathematics.

However, the apparent simplicity and accessibility of non-derivative methods do not necessarily extend to the associated mathematical analysis, as we shall see.

What's the status of non-derivative methods today???

Non-derivative methods have experienced a major renaissance, returning to grace, favor, and the interest of researchers.

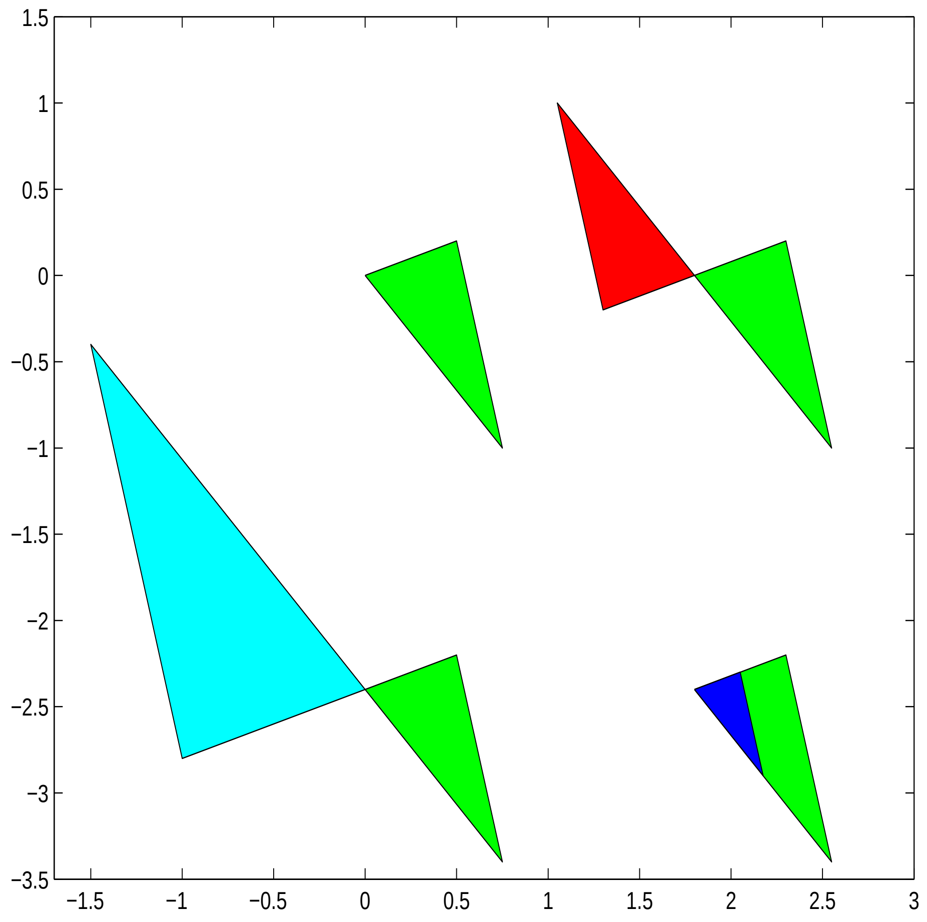
Since 1997, non-derivative methods, both model- and geometry-based, have been presented at mainstream optimization meetings.

The May 2008 SIAM Optimization Meeting included five sessions on non-derivative methods, and there will be a full "track" on this topic at the August 2009 International Symposium on Mathematical Programming.

One of the main factors in this sea change:

Torczon's (1989) PhD thesis presented a new geometry-based method (multidirectional search) with a convergence proof!!

Multidirectional search moves:



Goal: find a point strictly better than the *best* vertex.

After Torczon's thesis, a path of generalization (which mathematicians love) began and continues today:

Multidirectional search

(Torczon, 1989)

$\Downarrow$

Pattern search

(e.g., Torczon, 1997)

$\Downarrow$

Generalized pattern search

(Audet and Dennis, 2000)

An important definition in generalized pattern search methods: A set of vectors in $\mathcal{R}^n$

$$\mathcal{D} \triangleq \{d_\ell\}, \quad \ell = 1, \ldots, K,$$

is a **positive spanning set** for $\mathcal{R}^n$ if every $v \in \mathcal{R}^n$ can be written as a *nonnegative* linear combination of the elements of $\mathcal{D}$.

A positive spanning set must contain at least $n + 1$ vectors.

Standard examples in $\mathcal{R}^2$ of positive spanning sets:

$$\mathcal{D} = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix}$$

(Think coordinate search.)

$$\mathcal{D} = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix}$$

Each iteration of a generalized pattern search method includes an (optional) search step and a mandatory polling step.

The search step (which can be empty) is often based on properties of the particular problem being solved.

Convergence proofs depend on the polling step.

Excellent survey paper by Kolda, Lewis, Torczon, *SIAM Review*, 2003.

In the polling step at iteration k, x_k is the best point found so far. A set of directions $\mathcal{D}_k$ is chosen whose columns form a positive spanning set. (This set defines the "pattern".)

The function f is then evaluated at some, possibly all, of the "mesh-neighboring" points

$$\tilde{x}^{(\ell)} = x_k + \Delta_k d^{(\ell)}, \qquad d^{(\ell)} \in \mathcal{D}_k,$$

where Δ_k is a step control parameter.

Using a suitable acceptance strategy, if $\tilde{x}^{(\ell)}$ is strictly better than x_k, then $x_{k+1} \leftarrow \tilde{x}^{(\ell)}$ and $k \leftarrow k+1$.

If none of the sampled points is strictly better than x_k, x_k is said to be mesh-locally optimal, Δ_k is reduced in a controlled fashion, and the process is repeated.

Two points to note:

1. Most modern geometry-based methods no longer include an associated geometric figure (as does Nelder–Mead), so they are not so intuitively appealing to users.
2. The two most common variants of the acceptance strategy for a new best point are:
   - "Opportunistic" (accept a strictly better point as soon as you find it), or else
   - More demanding criteria (e.g., examine the complete set of possible new points and choose the best)

What kind of convergence results are known?

Theorem. (Torczon) If f is continuously differentiable and bounded below, then, for an opportunistic generalized pattern search algorithm,

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

NB: this theorem does **NOT** guarantee that the sequence of iterates converges to a stationary point.

Under “stronger hypotheses” (typically requiring more function evaluations),

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

There are convergence proofs (sometimes with “convergence” defined in a very limited sense) for the following categories of non-derivative methods, and more:

- pattern search and generalized pattern search,
- generating set search,
- adaptive pattern search,
- mesh-adaptive direct search,
- frame-based methods,
- grid-restrained methods,
- . . .

The proof techniques are very closely related to those in derivative-based optimization.

Having convergence proofs is important and comforting. But

- The results can be misunderstood to mean more than they do, e.g., $\liminf$ is not the same as $\lim$, nor is convergence of a subsequence that may or may not exist, and that can be identified only after the fact, the same as what most people view as “convergence”.

- The assumptions needed to prove convergence can be very far removed from the nasty properties of the functions we’re interested in.

- Unexpected things can happen even for apparently well-behaved functions.

Audet (2002) presents an array of examples that probe the limits of theory about generalized pattern search.

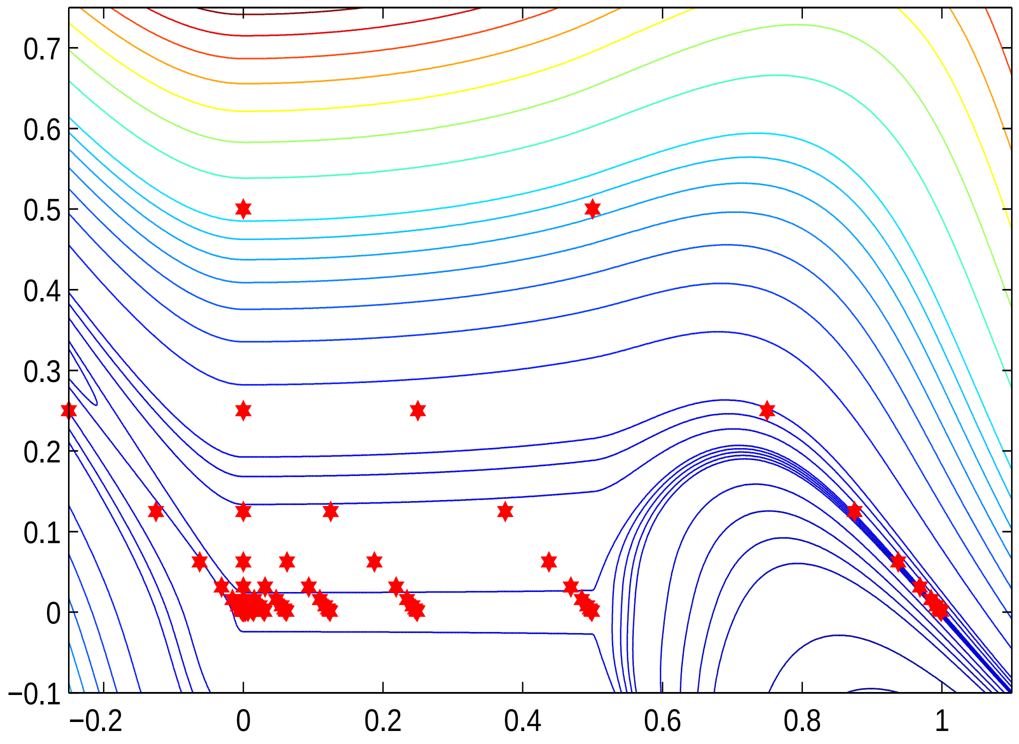
A continuously differentiable function of two variables:

if $x < 0$ then $f(x, y) = -26x^3 - 32x^2y + 7|y|^3$

else if $0 \leq x \leq \frac{1}{2}$ then $f(x, y) = (7 - 8x^2)|y|^3$

else $f(x, y) = (7 - 8x^2)|y|^3 + 8(x - \frac{1}{2})^2(y^3 + y + x - 1)$.

A specific generalized pattern search method applied to this function has an infinite number of limit points of the form $(1/2^\ell, 0)$, one of which—$(1,0)$—is not a stationary point.



So things are not always nice even when f is continuously differentiable.

Suppose next that f is arguably "nice" in some sense, but not continuously differentiable?

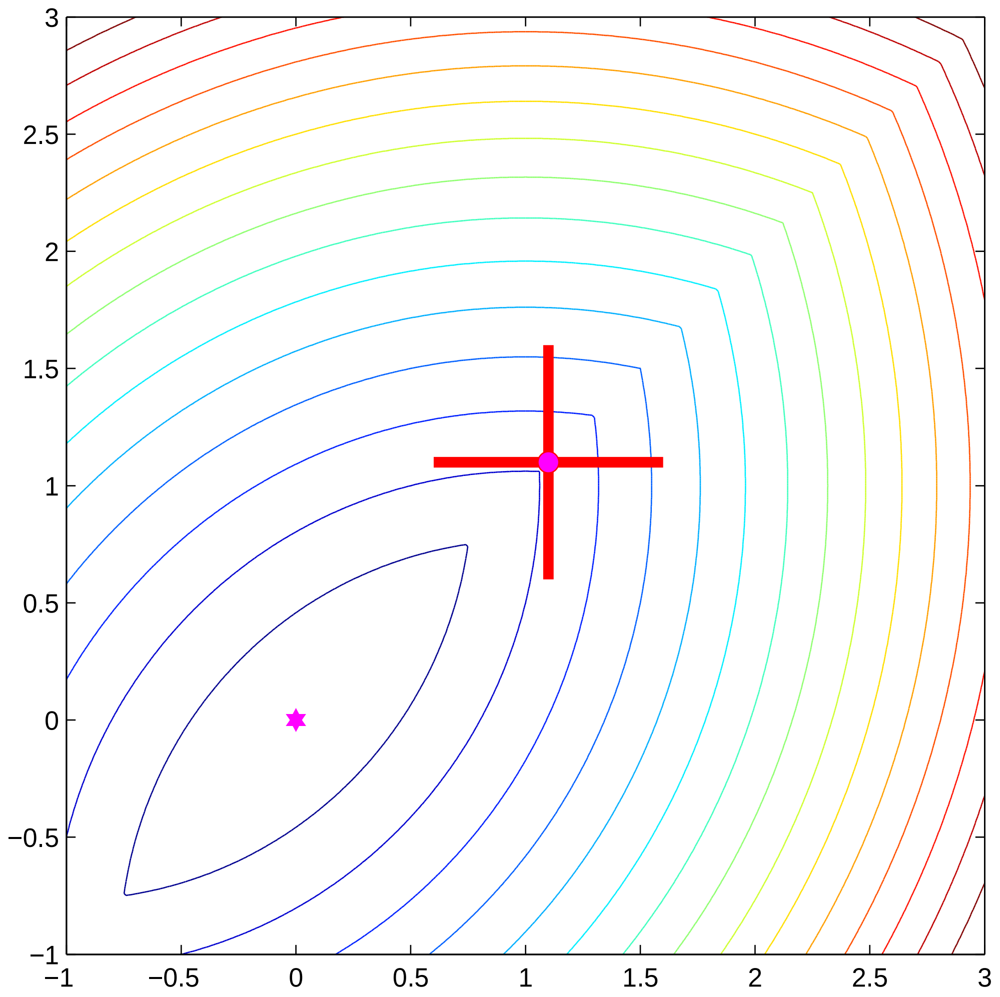
Modified Dennis–Woods function:

$$f_{DW} = \max(\|x - c\|^2, \|x - d\|^2)$$

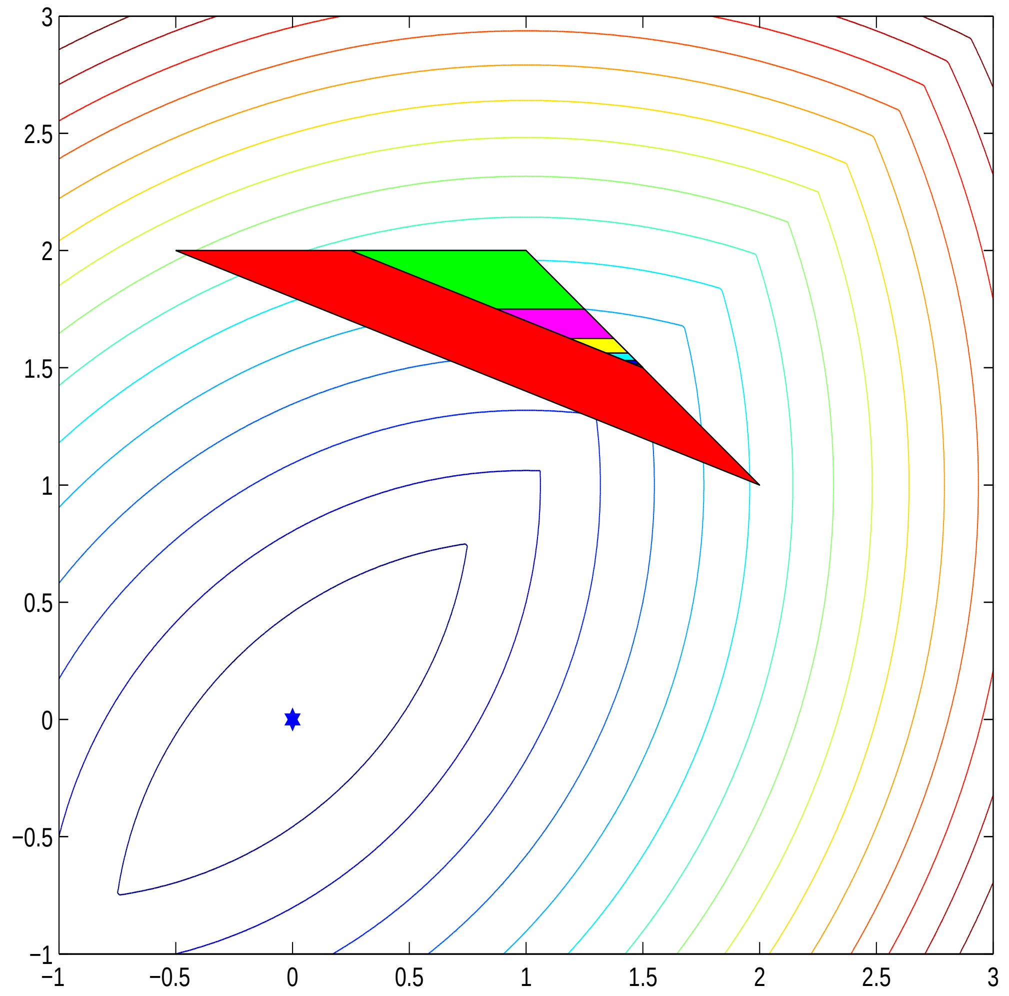
$$c = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \quad d = \begin{pmatrix} -1 \\ 1 \end{pmatrix}.$$

The gradient of f_{DW} is discontinuous along the line $x_1 = x_2$, and this seemingly minor nondifferentiability can cause difficulties for generalized pattern search methods.

Coordinate search on modified Dennis–Woods:



Multidirectional search on modified Dennis–Woods:



Turning now to something completely different. . . How does Nelder-Mead fit into the picture of generalized pattern search and related methods?

Not at all!

Among geometry-based non-derivative methods, the Nelder-Mead method is a far outlier (perhaps a singularity) in several ways.

- Each move is determined solely by the ordering of function values at the simplex vertices.
- A positive spanning set does not appear in the algorithm description.
- There is no "quality control" on the vertices or properties of the generated simplices.

Despite NM's apparent simplicity (one of its appeals to practitioners), it has proved to be very difficult to analyze mathematically, at least for those who have tried so far.

Part of the reason is that its algorithmic strategies are unusual (to say the least):

- it seeks to improve the worst point rather than the best; and

- the reflection point is accepted only if it is better than the second-worst vertex.

A few results from LRWW (1998):

1. In any dimension, Nelder–Mead is affine invariant.

2. In any dimension, with exact arithmetic, the NM simplex remains nondegenerate if the first simplex is nondegenerate.

3. In any dimension, if f is strictly convex, NM will take no shrink steps.

For strictly convex functions with bounded level sets:

Theorem. In dimension 1, NM converges to the minimizer.

Theorem. In dimension 2, the vertex function values converge in standard Nelder–Mead.

Theorem. In dimension 2, the simplex diameters converge to zero for standard Nelder–Mead.

NB: The third result does *not* prove convergence!

Points to note:

1. The LRWW proofs are based on treating the Nelder–Mead method as a discrete dynamical system, a non-standard approach in analysis of optimization methods.

2. The restriction to two dimensions arises because certain properties needed in the LRWW proofs do not hold if $n > 2$:

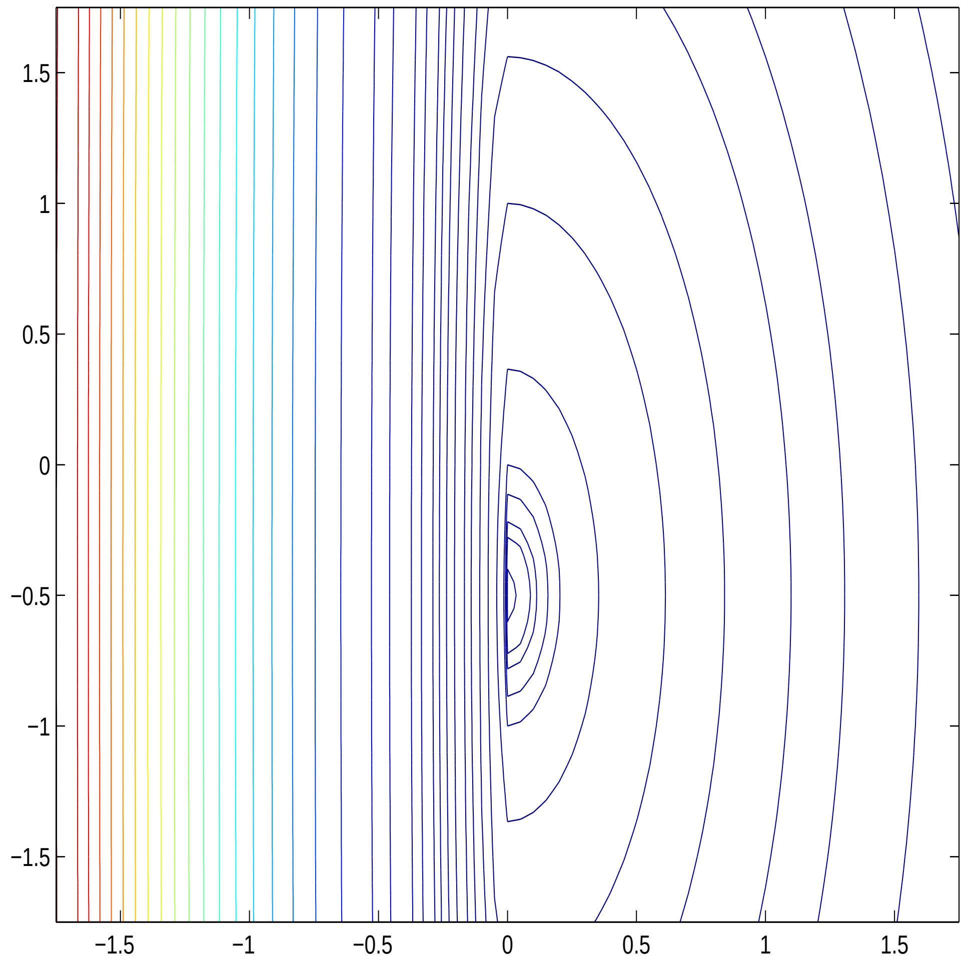
   - A reflected triangle is congruent in 2-d, but not for $n \geq 3$.

   - There are explicit forbidden move sequences in 2-d (e.g, five consecutive reflections in which the next-worst point changes), but no such sequences are known for $n \geq 3$.

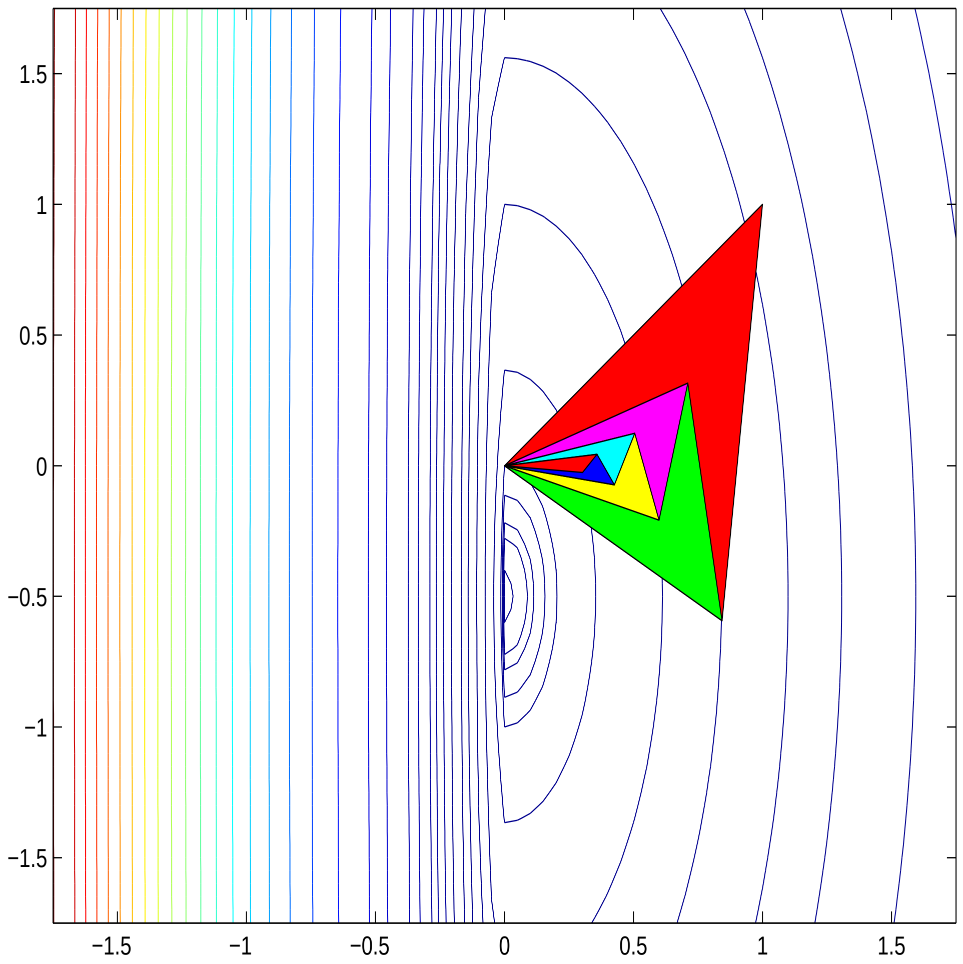
And even in 2-d, there is a counterexample, a strictly convex function with bounded level sets, for which Nelder–Mead converges to a nonminimizer, never changing the best vertex of the original simplex (McKinnon, 1998).

The failure involves an infinite sequence of inside contractions.

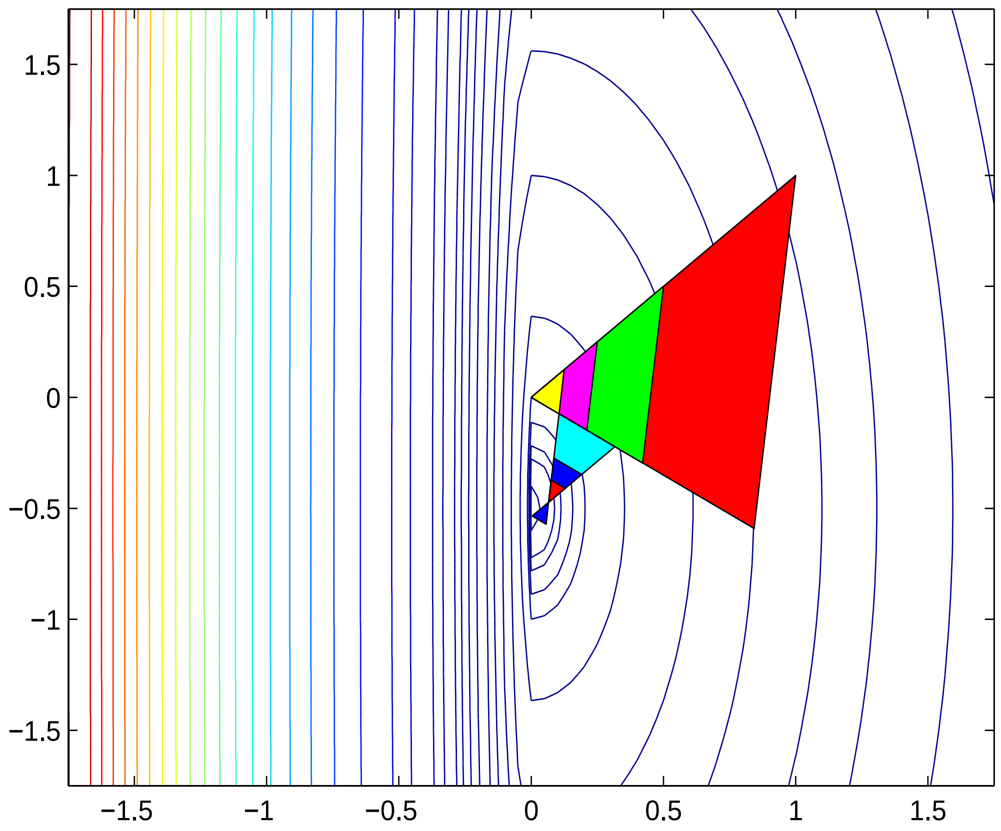
The McKinnon counterexample



Nelder-Mead on the McKinnon counterexample



Multidirectional search on the McKinnon counterexample



A recent new theoretical result about Nelder–Mead:

In dimension 2, for functions

> that are strictly convex with bounded level sets,
>
> that are twice continuously differentiable and
>
> whose Hessians are uniformly bounded away from singularity. . .

● ● ●● ●● ● ●

Theorem. (LPW, 2009) The restricted Nelder–Mead method (*with no expansion steps*) converges to the minimizer.

NB: The smoothest McKinnon example does not satisfy these more limited conditions, since the matrix of second partial derivatives is singular at the vertex that causes the failure.

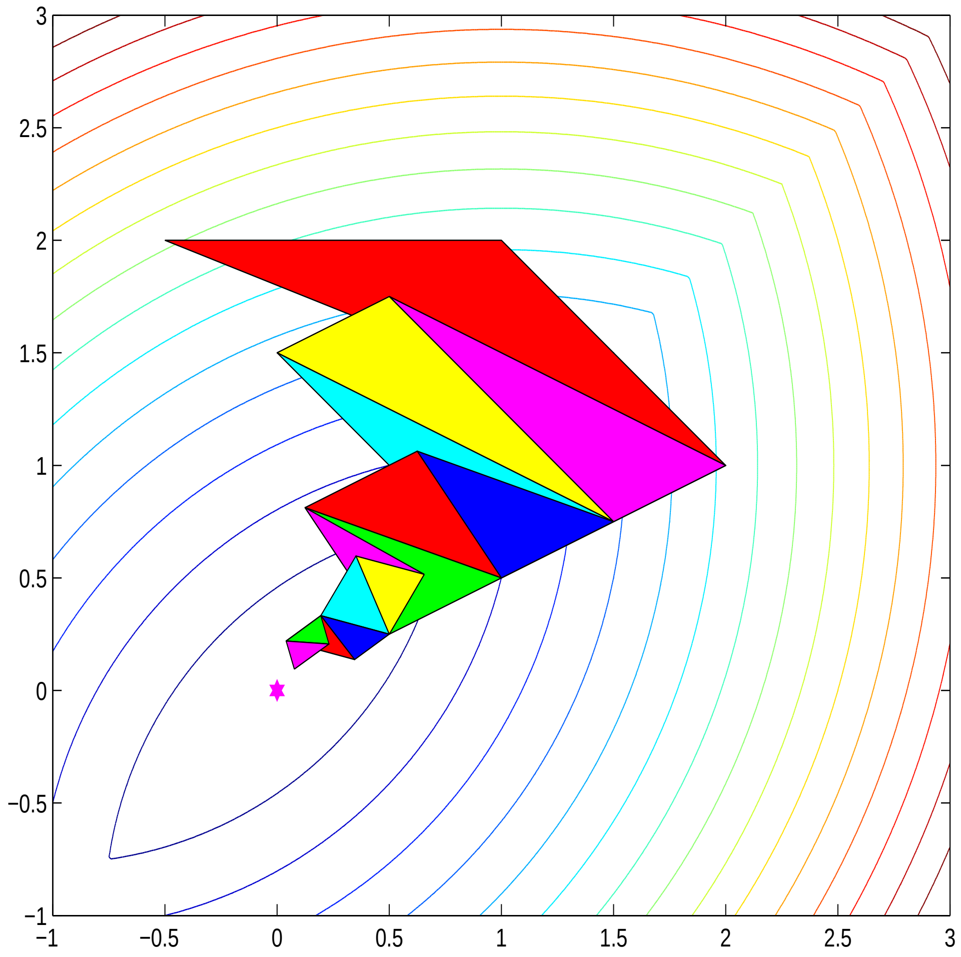
Given its extremely limited theory, no general convergence proof even in 2-d, the McKinnon counterexample. . .

Why not just abandon Nelder–Mead??

Because often it works (and works well) when other methods don't.

But we still don't know why!

Example: on Dennis-Woods, where pattern search methods fail. . .



Mathematical results that would be useful about Nelder–Mead:

Does it converge under the assumptions of the LRWW paper?

Is there a counterexample in three dimensions à la McKinnon for the restricted Nelder–Mead method under the stricter assumptions of the 2009 LPW result?

What is the theoretical (and/or likely) effect of problem dimension? (There is longstanding evidence that efficiency deteriorates with dimension for non-derivative methods.)

Limited results about the effect of dimensionality are shown in L. Han and M. Neumann (2006), *Effect of dimensionality on the Nelder-Mead simplex method*.

Under several assumptions (discussed in a moment), they show that convergence of the Nelder-Mead simplex to zero is linear, with an asymptotic error constant that rapidly approaches 1 as n increases, showing that performance deteriorates with dimension.

The analysis is *very interesting*, and the conclusion matches longstanding folklore.

But it answers only some of the open questions about Nelder–Mead and problem dimension.

In the Han-Neumann analysis,

- $f = x^T x$ (so general quadratics are covered, using affine-invariance);
- One of the vertices of the initial simplex is at the minimizer;
- The moves analyzed involve infinite sequences of one move, e.g., inside contraction (as in McKinnon's example).

Can a similar analysis be done that would apply when started with a general nondegenerate simplex?

And what goes wrong in practice with NM?

- The simplex becomes close to degenerate, i.e., "almost collapses into a subspace", despite the theoretical guarantee that it will remain nondegenerate.

  (But this can be good, not bad, when the simplex is "elongating" along directions that follow the function.)

- The simplex "stagnates", or the optimization "stalls", and this can happen even when the simplex does not seem to be close to degenerate.

Needed: a more complete analysis of failure modes.

A complication in figuring this out: some reported failures happen only after thousands of iterations, and only for specific starting simplices.

For geometry-based methods, there is a major open question that is both mathematical and practical. . .

Getting a better answer quickly is often what users really want, especially when optimizing models that are known to be inaccurate.

There's significant computational evidence that Nelder–Mead, despite its lack of theory, is often (but not always) better in this regard than generalized pattern search methods.

But we don't really know how to define "getting a better answer quickly" in a rigorous sense that can be analyzed mathematically.

A (heuristic) approach: try to define classes of functions for which different methods work more or less well, and explain why.

Conclusions:

There are many open and interesting mathematical and computational questions about non-derivative optimization methods.

We need

- mathematics that helps us to understand the theoretical properties as well as the performance in practice of non-derivative methods, and
- clever numerical algorithms, possibly based on sound heuristics, that will solve important real-world problems.