Unconstrained Minimization in $\mathbb{R}^n$

Direct Search Methods
(nongradient methods)

1. Random search methods
2. Univariate method (one variable at a time)
3. Pattern search methods
   a) Hooke and Jeeves method
   b) Powell's conjugate direction method
4. Rosenbrock's method of rotating coordinates
5. Simplex method

Descent Methods
(gradient methods)

1. Steepest descent method
2. Conjugate gradient method (Fletcher - Reeves)
3. Newton's method
4. Variable metric method (Davidon - Fletcher - Powell)

- All of the above methods are iterative type methods which start with a trial solution x_i in n-dimensional space and proceed in a sequential manner. The general procedure can be written as follows:

Algorithm: General n-dimensional search iteration

1. Input x_1
2. set $i = 1, F_i = F(x_i)$
3. repeat
4. set $i = i + 1$
5. Generate new point x_i
6. set $F_i = F(x_i)$
7. continue until some convergence criterion is met
8. stop

We will look at three techniques which do not require derivative information

Heuristic techniques

1. Simplex Search (S^2 method) (Nelder - Mead)
2. Hooke -Jeeves Pattern Search

Theory based Technique

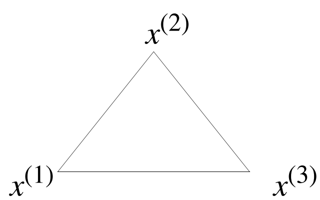
3. Powell's Conjugate Direction Method

Simplex Search or s^2 Method

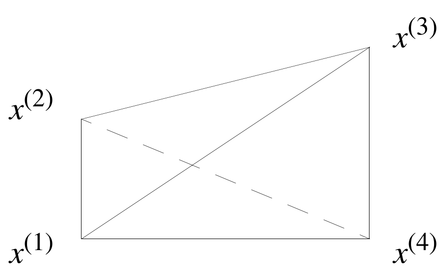
Definition: a *simplex* is a geometric figure formed by n+1 points in an n-dimensional space. If the points are equidistant from each other then it is called a *regular simplex*. ❐

Examples

1) an equilateral triangle is a regular simplex in 2-dimensional space (3 - points).



2) the tetrahedron shown in the figure is a simplex in 3-dimensional space (4 - points)



3) a polyhedron composed of n+1 equidistant points in n-dimensional space is a regular simplex.

Algebraic method of constructing a simplex

- Consider a *base point*, given as

$$x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)})$$

then a regular simplex having sides of length α can be defined as

$$x_j^{(i)} = \begin{cases} x_j^{(0)} + \delta_1, i = j \\ x_j^{(0)} + \delta_2, i \neq j \end{cases}, \quad i, j = 1(1)n$$

where

$$\delta_1 = \alpha\left[\frac{(n+1)^{1/2} + n - 1}{n\sqrt{2}}\right],$$

$$\delta_2 = \alpha\left[\frac{(n+1)^{1/2} - 1}{n\sqrt{2}}\right]$$

where α is chosen by the user for the length of the sides of the simplex.

Example

- Consider the following 2-dimensional example: the base point is the origin - (0,0), $\alpha = 1$ and of course $n = 2$. Thus $x^{(0)} = (0,0)$ and we have

$$\delta_1 = \left[\frac{\sqrt{3}+1}{2\sqrt{2}}\right] = 0.966, \; \delta_2 = \left[\frac{\sqrt{3}-1}{2\sqrt{2}}\right] = 0.259$$

$$x_1^{(1)} = 0 + 0.966 = 0.966, \; x_2^{(1)} = 0 + 0.259 = 0.259$$

$$x^{(1)} = (0.966, 0.259), \; x_1^{(2)} = 0 + 0.259 = 0.259, \; x_2^{(2)} = 0 + 0.966 = 0.966$$

$$x^{(2)} = (0.259, 0.966)$$

We can check that the sides of the simplex have length $\alpha = 1$:

$$\|x^{(1)} - x^{(2)}\| = \sqrt{(0.966 - 0.259)^2 + (0.259 - 0.966)^2} = 1.0$$

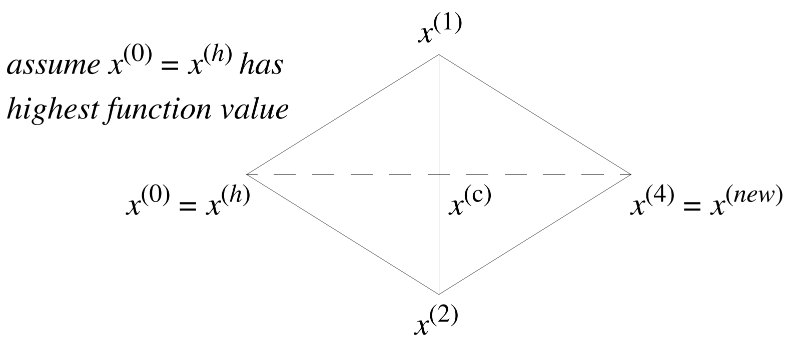
$$\|x^{(0)} - x^{(1)}\| = \sqrt{(0.966)^2 + (0.259)^2} = 1.0$$

$$\|x^{(0)} - x^{(2)}\| = \sqrt{(0.259)^2 + (0.966)^2} = 1.0 . \blacksquare$$

- Now how do we use the simplex in an algorithm to search for the minimum of a multivariate function?
- An original method developed by **Spendley, Hext and Himsworth**, in **1962** uses the following constructions to move a simplex towards a minimum.

1) Reflection through the centroid (standard simplex)

The objective function is evaluated at all $n + 1$ vertices of the simplex and the vertex of highest function value is *reflected through the centroid of the opposite side face on the simplex* as shown below for the 2-dimensional case.



Reflection of vertex with highest function evaluation through centroid.

- How do we perform this reflection algebraically? (Certainly in n- D we can't imagine this.)
- Algebraically, suppose

$$x^{(j)} = x^{(h)}$$

is the point with the highest evaluated function value which must be reflected. The centroid of the N remaining points is given by

$$x^{(c)} = \frac{1}{n} \sum_{i = 0, i \neq h}^{n} x^{(i)}$$

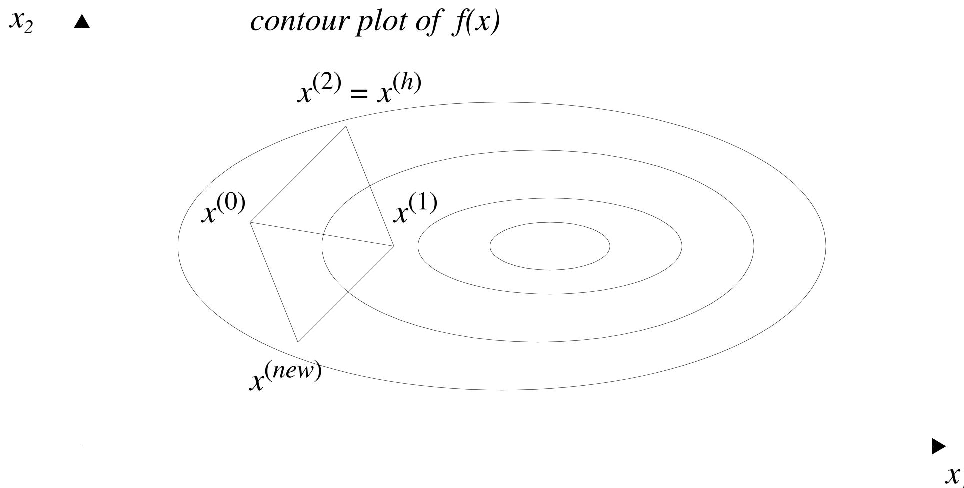
and the line through the centroid and the point with highest function value, $(x^{(c)}, x^{(h)})$, is given by the vector form of a line

$$x = x^{(h)} + \lambda(x^{(c)} - x^{(h)})$$

where λ is a scaling parameter. In this equation, if $\lambda = 0 \Rightarrow x = x^{(h)}$, if $\lambda = 1 \Rightarrow x = x^{(c)}$, and if $\lambda = 2 \Rightarrow x^{(new)} = x = x^{(h)} + 2(x^{(c)} - x^{(h)}) = 2x^{(c)} - x^{(h)}$.

- This procedure of continually reflecting the vertex with the highest function value will generally move the simplex towards the minimum of the function but has the following difficulties or problems:

1. if $f(x^{(new)}) \geq f(x^{(h)})$ then we will just reflect back and forth. This may happen when we're stuck in a trough as shown in the figure.



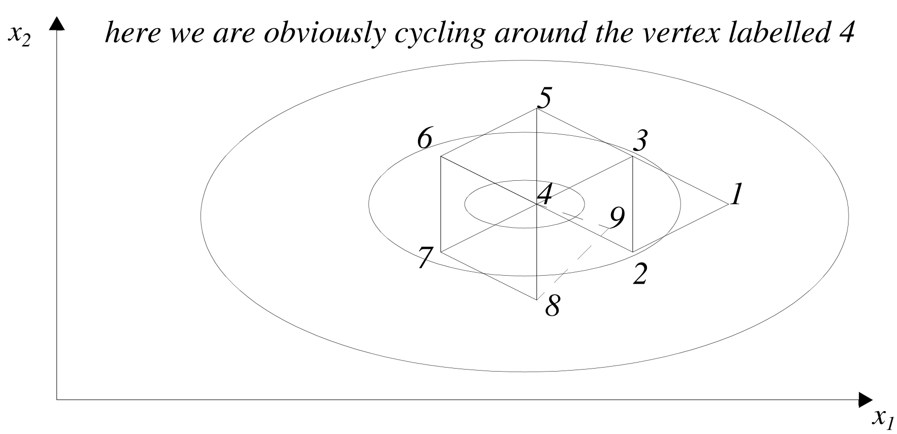
Simplex stuck in a trough, $x^{(new)}$ is highest point.

One possible solution to this problem is to use the next highest point as the point we will reflect. Therefore in since $f(x^{(new)}) \geq f(x^{(h)})$ we would reflect $x^{(0)}$, that is the next highest point, on the second iteration.

2. Even with the above modification sometimes one vertex remains unchanged for more than M iterations and we are cycling around one point because our simplex is too large.
One possible solution to this problem is to set up a new simplex with the lowest point as the base point and reduce α (*i.e.* the simplex size). How do we know when to do this? A heuristic formula for the number of iterations M is given by

$$M = 1.65n + 0.05n^2$$

where M is rounded to nearest integer (here, n is the problem dimension).



Nelder - Mead modifications of the simplex routine

- The original simplex method was later developed more fully by Nelder and Mead in 1965 to take into account some problems which may occur.

- Now contraction and expansion of the simplex is allowed, *i.e.* it no longer has to remain a regular simplex. The line of reflection is now written as:

$$x = x^{(h)} + (1+\theta)[x^{(c)} - x^{(h)}]$$

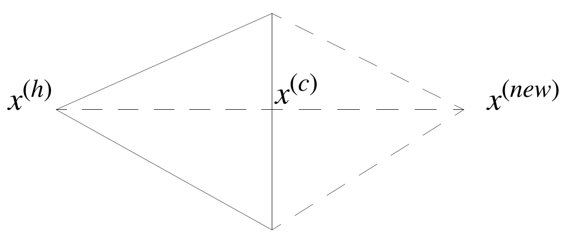
and we consider the point with the next highest current function value and the lowest current function value:

$x^{(g)}, f^{(g)}$ - next highest current point

$x^{(l)}, f^{(l)}$ - lowest current point

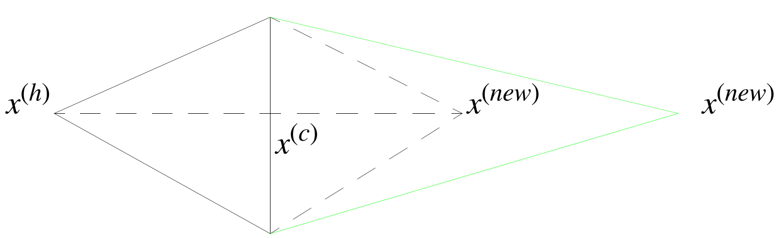
The possible types of reflections are now determined as follows:

(a) <u>normal reflection</u>: if $f^{(l)} < f^{(new)} < f^{(g)}$, choose: $\theta = \alpha = 1$.



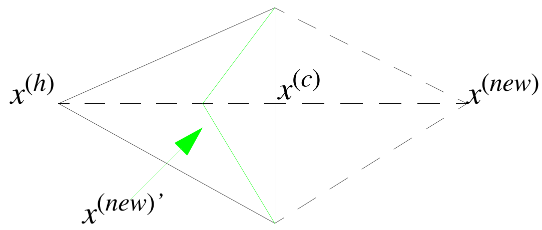
Normal reflection as in the standard simplex method.

(b) <u>expansion</u>: if $f^{(new)} < f^{(l)}$ then the new point is less than even the lowest point so take advantage and move even more in that direction. Choose: $\theta = \gamma > 1$.



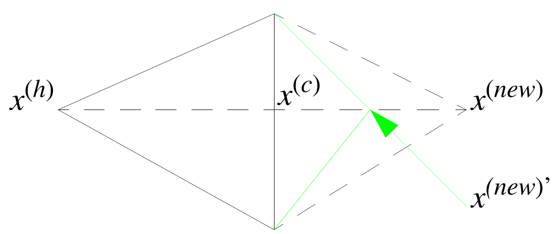
Expansion.

(c) <u>contraction 1</u>: if $f^{(new)} > f^{(h)}$ then we must be in a trough so move $x^{(h)}$ in a bit to see what happens. Choose: $\theta = \beta < 0$.



Contraction 1.

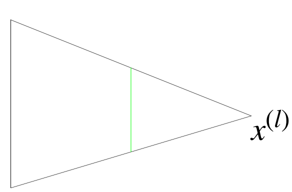
(d) <u>contraction 2</u>: if $f^{(g)} < f^{(new)} < f^{(h)}$ then the new point is lower but not by much. Choose: $\theta = \beta$, $0 < \beta < 1$.



Contraction 2.

Some "<u>recommended</u>" values are: $\alpha = 1$, $\beta = \pm 0.5$, $\gamma = 2$

(e) <u>contraction 3</u>: if all the above procedures fail to produce a lower point than at $x^{(l)}$ after M tries then contract all sides towards $x^{(l)}$. [see Numerical Recipes: amoeba ( )].



Contract along all dimensions towards the lowest point.

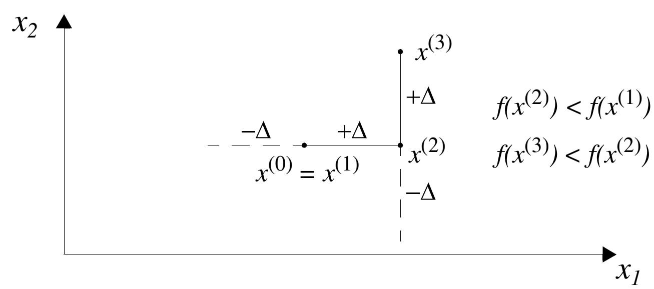
The Hooke and Jeeves Direct Search

The *Hooke-Jeeves* method is a heuristic direct search technique which uses “exploratory moves” to find a good direction and then conducts a “pattern move” in that direction. These are called moves as opposed to minimizations since fixed steps are taken in certain directions.

Exploratory Moves

- Given an initial starting point called a “base point” $x^{(0)}$ in n-dimensional space we set the initial “search point” $x^{(1)}$ to $x^{(0)}$. That is $x^{(1)} = x^{(0)}$.

- Then using n orthogonal unit vectors supplied by the user, which are usually the coordinate directions: $e^{(i)}, i = 1(1)n$, we search a $\underline{\pm\Delta}$ amount in each direction, where Δ is initially input, for a lower function value. After these n searches we arrive at the point $x^{(n+1)}$.



Exploratory moves.

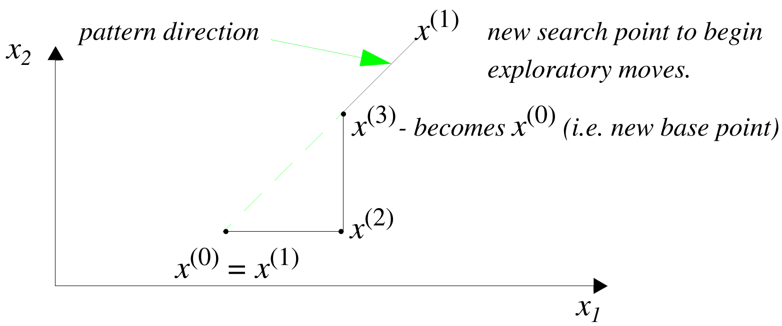
Pattern Move

1. if the final point x^{n+1} of the exploratory moves has a lesser function value than the base point $x^{(0)}$, that is if $\underline{f(x^{(n+1)}) < f(x^{(0)})}$, then make the new search point equal to

$$x^{(1)} = x^{(n+1)} + (x^{(n+1)} - x^{(0)})$$

   and make the new base point $x^{(0)} = x^{(n+1)}$.

   (The direction $x^{(n+1)} - x^{(0)}$ is referred to as the pattern direction.)



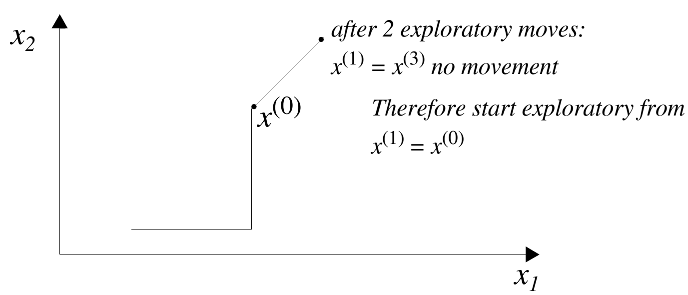
Pattern direction move.

2. if case 1) is not true and the base point and search point are the same, that is $(x^{(1)} \equiv x^{(0)})$, then reduce the step amount, say $\Delta = \Delta / 10$ (the reduction factor 10 is input).

3. if case 1) is not true but base point and search point are not equal then make them the same,

$$x^{(1)} = x^{(0)}$$

and go to back to exploratory moves.



No movement after exploratory moves, therefore go back to base point.

Algorithm: Hooke and Jeeves Direct Search

1. input $\Delta, x^{(0)}, \delta, e^{(i)}, i = 1, \ldots, n$
2. set $x^{(1)} = x^{(0)}$
3. repeat
4. for $i = 1, 2, \ldots, n$
5. set $x = x^{(i)} + \Delta e^{(i)}$
6. if $f(x) < f(x^{(i)})$
7. set $x^{(i+1)} = x$
8. else
9. set $x = x^{(i)} - \Delta e^{(i)}$
10. if $f(x) < f(x^{(i)})$
11. set $x^{(i+1)} = x$
12. else
13. set $x^{(i+1)} = x^{(i)}$
14. end
15. end
16. end
17. if $f(x^{(n+1)}) < f(x^{(0)})$
18. set $x^{(1)} = x^{(n+1)} + (x^{(n+1)} - x^{(0)})$
19. $x^{(0)} = x^{(n+1)}$
20. else if $x^{(1)} = x^{(0)}$
21. set $\Delta = \frac{\Delta}{10}$
22. else
23. set $x^{(1)} = x^{(0)}$
24. end
25. end
26. continue until $\Delta < \delta$

- In the Hooke-Jeeves algorithm, lines 4 - 16 represent the exploratory moves while lines 17 - 25 represent the pattern moves. The algorithm terminates when the step size is less than a user specified tolerance size δ.

Minimization of a Multivariable Function Along a Line

- Thus far we have discussed heuristic methods where we have taken discrete "steps" in chosen directions but the size of the steps was determined heuristically.

Question: If we have a chosen direction in n-dimensional space, how do we minimize in that direction?

- That is, given $f(x), x_0, u$ where $x \in \mathbb{R}^n$, x_0 is the starting point, and $u \in \mathbb{R}^n$ is the direction in which we want to minimize $f(x)$, how do we perform this line minimization?
- Starting from x_0 we look for a new point

$$x = x_0 + \lambda^* u$$

such that $F(\lambda) = f(x_0 + \lambda u)$ is minimized when $\lambda = \lambda^*$. We treat $F = F(\lambda)$ as a function of only λ (*i.e.* of only one variable).

EXAMPLE

Consider the function $f(x) = 1 + x_1 - x_2 + x_1^2 + 2x_2^2$.

Minimize in the $u = \begin{pmatrix} -2 \\ 1 \end{pmatrix}$ direction starting from $x_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.

<u>Solution</u>:

We first form the line to search on:

$$x = x_0 + \lambda u = \begin{pmatrix} -2\lambda \\ \lambda \end{pmatrix}$$

and then the function of only one variable

$$f(x_0 + \lambda u) = 1 - 2\lambda - \lambda + 4\lambda^2 + 2\lambda^2 = 1 - 3\lambda + 6\lambda^2 = F(\lambda)$$

which can be minimized by

$$\left.\frac{dF}{d\lambda}\right|_{\lambda^*} = -3 + 12\lambda^* = 0 \Rightarrow \lambda^* = \frac{1}{4}.$$

By the second derivative we see that

$$\left(\frac{d^2F}{d\lambda^2} = 12 > 0\right) \Rightarrow \lambda^* = \frac{1}{4} \text{ is the minimum.}$$

Therefore

$$x^* = \begin{pmatrix} -1/2 \\ 1/4 \end{pmatrix}$$

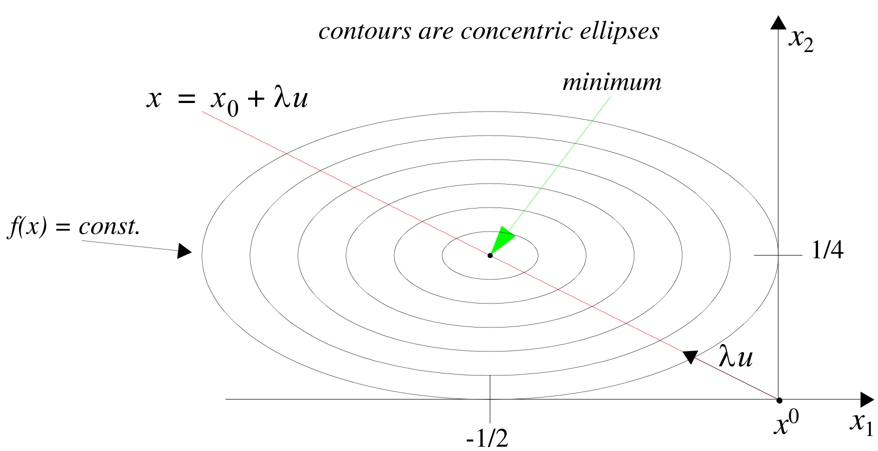
is the minimum of $f(x)$ along the $x = x_0 + \lambda u$ line.

In fact we can show that this is the global minimum of $f(x)$ (by chance). Completing the square, we can write $f(x)$ as

$$f(x) = \left(x_1 + \frac{1}{2}\right)^2 + 2\left(x_2 - \frac{1}{4}\right)^2 + \frac{5}{8}$$

which has contours consisting of concentric ellipses centered about its minimum at $x^* = (-1/2, 1/4)$. As can be seen in , the minimum along the line $x = x_0 + \lambda u$ passes through the minimum.

The minimum of $f(x)$ is $f(x^*) = 5/8$.



Minimum along a line turns out by chance to be the minimum of the function.