Thus, $\lim_{k \to +\infty} \| \nabla f(x_{\varphi(k)}) \| = 0$

$\Longrightarrow$ More details for the proof of the lemma: PhD thesis B. Pauwels (2016) or:

[61] Kolda, T. G., Lewis, R. M., and Torczon, V. Optimization by direct search : New perspectives on some classical and modern methods. *SIAM review 45*, 3 (2003), 385–482.

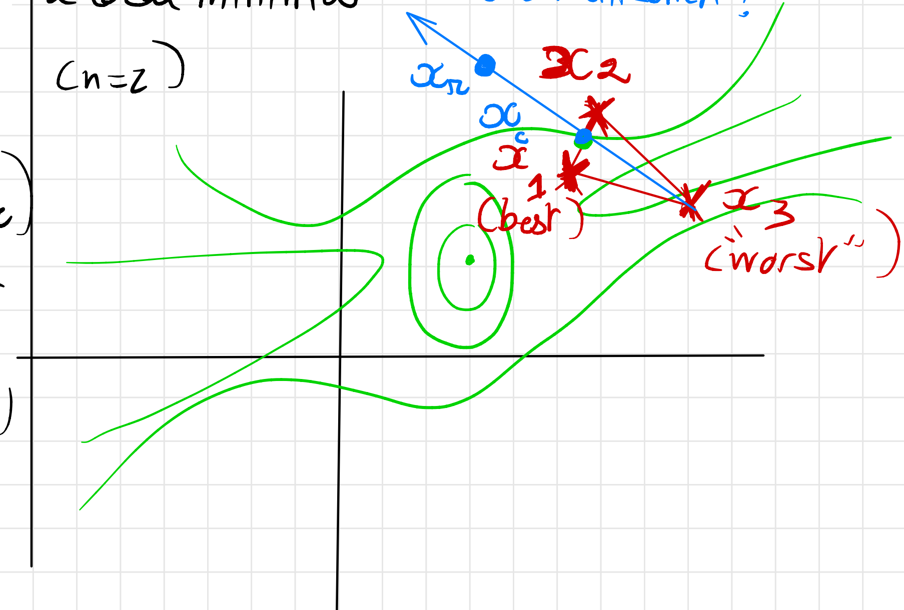
In the next session, we will present a new approach (simplex-type) for local optimization, in particular the Nelder-Mead algorithm (1965, still used in Matlab).

2.2. Nelder Mead method

Idea (1965): make evolve a set of $(n+1)$ linearly independant points (called a simplex) in order to reach a local minima

descent direction ?

$(n=2)$



The general algorithm is the following:

→ For a given simplex $\{x_1^k, x_2^k, \dots, x_{n+1}^k\}$ where

$$f(x_1^k) \leq f(x_2^k) \dots \leq f(x_{n+1}^k)$$

1) Compute

$$x_c = \frac{1}{n} \sum_{i=1}^{n} x_i^k$$

2) Compute $x_r = x_{n+1}^k + 2\overbrace{(x_c - x_{n+1}^k)}^{d}$

and evaluate $f(x_r)$

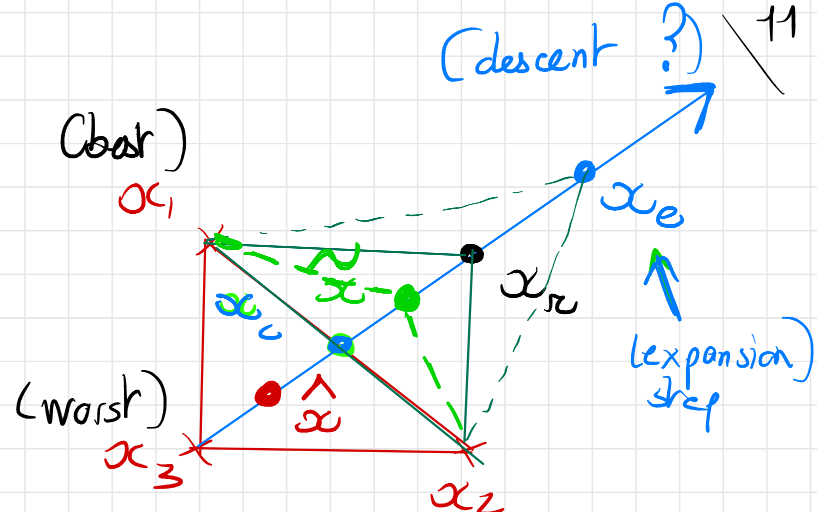
4 cases can occur

Case 1 : $f(x_r) < f(x_1^k)$

Case 2 : $f(x_1^k) \leq f(x_r) \leq f(x_n^k)$

Case 3 : $f(x_n^k) < f(x_r) \leq f(x_{n+1}^k)$

Case 4 : $f(x_{n+1}^k) < f(x_r)$



Case 1 : Define $x_e = x_{n+1}^k + 3(x_c - x_{n+1}^k)$ and evaluate $f(x_e)$.

→ Replace in the simplex, x_{n+1}^k by the best point between x_r and x_e.

Case 2 : Replace in the simplex x_{n+1}^k by x_r.

Case 3 : compute

$$\tilde{x} = x^k_{n+1} + 3/2\,(x_c - x^k_{n+1})$$

and evaluate $f(\tilde{x})$.

→ Replace in the simplex the point x^k_{n+1} by the best point between x_r and $\tilde{x}$.

Case 4 : compute

$$\hat{x} = x^k_{n+1} + \frac{1}{2}(x_c - x^k_{n+1})$$

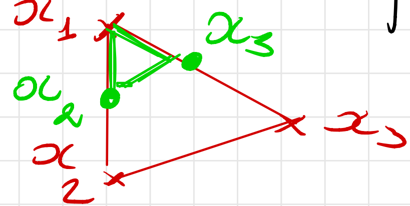
and evaluate $f(\hat{x})$.

If $f(\hat{x}) \leq f(x^k_{n+1})$ then replace x^k_{n+1} by $\hat{x}$ in the simplex.

Else (failure step) : apply a shrink step centered at x^k_1 :

$$x^{k+1}_j = x^k_1 + \frac{1}{2}(x^k_j - x^k_1)$$

and evaluate $f(x^{k+1}_j)$ $1 \leq j \leq n$



* Implementation (with Scilab) :

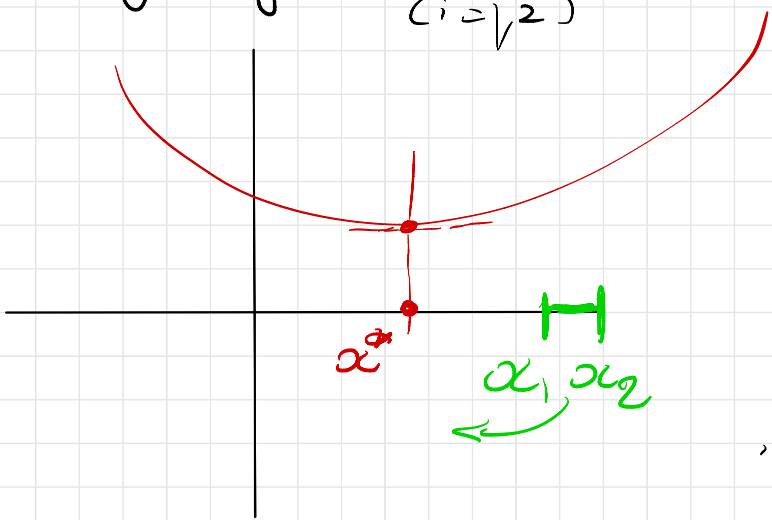
* Convergence to a local minimum of the Rastrigin function :

$$f(x_1, \dots, x_n) = n + \sum_{i=1}^{n} \left(x_i^2 - 2\cos(2\pi x_i)\right)$$

(function with a large number of local minima and a unique global one at $(0, .. 0)$)

* Some partial convergence results:

If $n=1$, and f is strictly convex and coercive, then the Nelder Mead algorithm converge to the unique global minimum of f (convergence of $x_i^k \to x^*$ $(i=1,2)$)



If $n=2$, and f is strictly convex and coercive, then:

$$\rightarrow \begin{cases} f_1^* = \lim\limits_{k\to+\infty} f(x_1^k) \\ f_2^* = \lim\limits_{k} f(x_2^k) \\ f_3^* = \lim\limits_{k} f(x_3^k) \end{cases}$$ exists and are equal.

Moreover $\operatorname{diam}(\{x_1^k, \dots x_3^k\}) \to 0$ (diameter of the simplex)

... but $\Delta_k = \{x_1^k, \dots x_3^k\}$ does not necessarily converge to x^*.

(1998, MacKinnon and al, SIAM J.)

(Counterexample: see implementation)

14

Even if a very specific non convergence result has be proven, this algorithm, remains very popular and can be used in many configurations (it is the default algorithm for the solver fminsearch in Matlab)

Some variants exists in order to achieve a theoritical convergence results (reference: DFO book).

Exercices:

Ex 2)

Exercice 2 *On the Nelder Mead algorithm*

1. Recall briefly the main principles of the Nelder Mead algorithm. A 2D illustration of the possible steps can be used.
2. Prove that no shrinkage steps are performed when the Nelder Mead algorithm is applied to a strictly convex function. We recall that $f : \mathbb{R}^n \to \mathbb{R}$ is strictly convex if and only if :

$$\forall (x,y) \in \mathbb{R}^n \times \mathbb{R}^n,\ \forall \lambda \in]0,1[,\ f(\lambda x + (1-\lambda)y) < \lambda f(x) + (1-\lambda) f(y) \text{ if } x \neq y$$

1) see above

2) Suppose that a shrinkage occurs:

Exercice 4 *On the Nelder Mead algorithm*

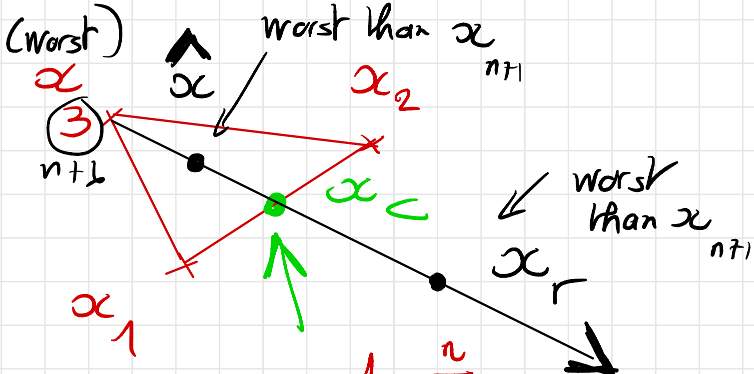
15

Consider the Nelder Mead algorithm for the sphere function in $\mathbb{R}^2$:

$$f(x,y) = x^2 + y^2$$

and the initial simplex made of $A = (4,5)$, $B = (5,3)$ and $C = (5,6)$.

Compute the best obtained value by this algorithm after one iteration.

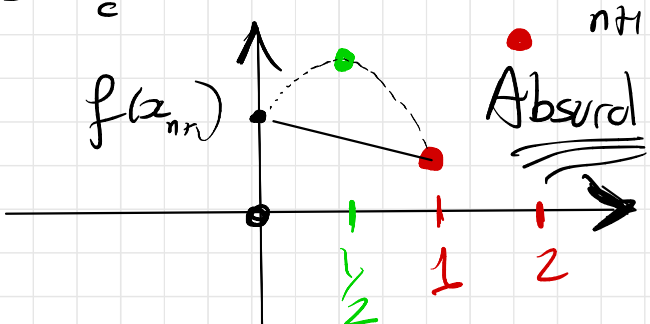


$$x_c = \frac{1}{n} \sum_{i=1}^{n} x_i$$

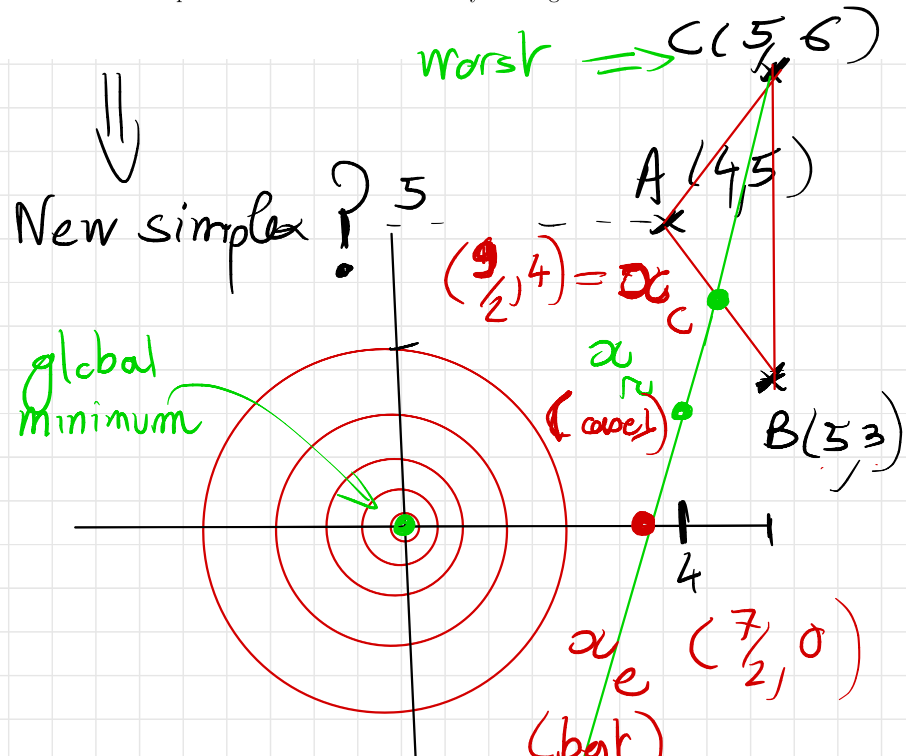
By convexity

$$f(x_c) \leq \frac{1}{n} \sum_{i=1}^{n} f(x_i)$$

$\Rightarrow$ x_c is "better" than x_{n+1}



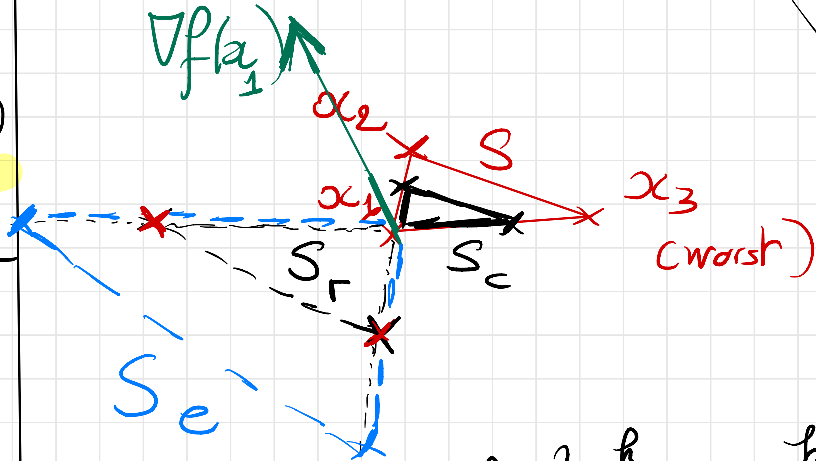
New simplex ?



C is replaced by $(7/2, 0) = x_e$

2.3) Multi Direction Search method (MDS)

* The idea is still to make evolve a simplex, but without changing its topology (which is not the case in the Nelder Mead approach and can lead to troubles).
The "price to pay" is to make more evaluations at each step.
(Virginia Torczon, PhD, 1989)



→ For a given simplex $\{x_1^k, \dots x_{n+1}^k\} = S$ such that
$f(x_1^k) \leq \dots \leq f(x_{n+1}^k)$

* Compute S_r, the reflected simplex of S with respect to x_1^k and evaluated $f(S_r)$, that is all the points of this new simplex.

Two cases can occur:

→ $\mathrm{Min}(f(S_r)) < f(x_1^k)$

Compute S_e by extrusion (factor 2) of S_r and evaluate $f(S_e)$

↗ If $\mathrm{Min}(f(S_e)) < \mathrm{Min}(f(S_r))$ then replace S by S_e

else replace S by S_r.

→ $\mathrm{Min}(f(S_r)) \geq f(x_1^k)$

Compute S_e by contraction (factor $\frac{1}{2}$) of S around x_1^k.

Replace S by S_e //

→ Implementation with Scilab

Convergence to a local (or global minima) for every initialisation and any cost function.

There exists a strong convergence result (proven in V. Torczon PhD).

Theorem: Assume f C^1 and $L(x_1^0) = \{x \in \mathbb{R}^n, f(x) \leq f(x_1^0)\}$ is compact. (for example, if f coercive).

There exists a subsequence of (x_1^k) that converges to a stationary point x_{dk}.

Corollary: Assume that f is strictly

convex and coercive, then

$x^k \to x^*$, global minimum of f.

The proof relies on the observation that the MDS algorithm behaves like a descent method, that does not degenerate, with a sufficient decrease compared to the stepsize.

Compared to Nelder Mead, the convergence result is better, but the implementation is more computationaly expansive:

Step $k \to$ Step $k+1$:

MDS needs $(2N)$ evaluations in all cases

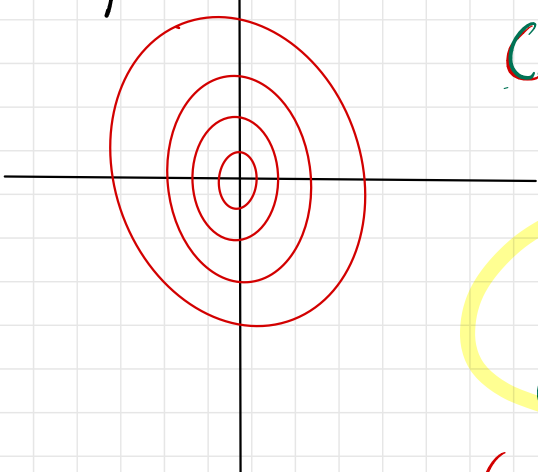
NM needs 2, or 1, or 2, or 2 or $n+2$ evaluations

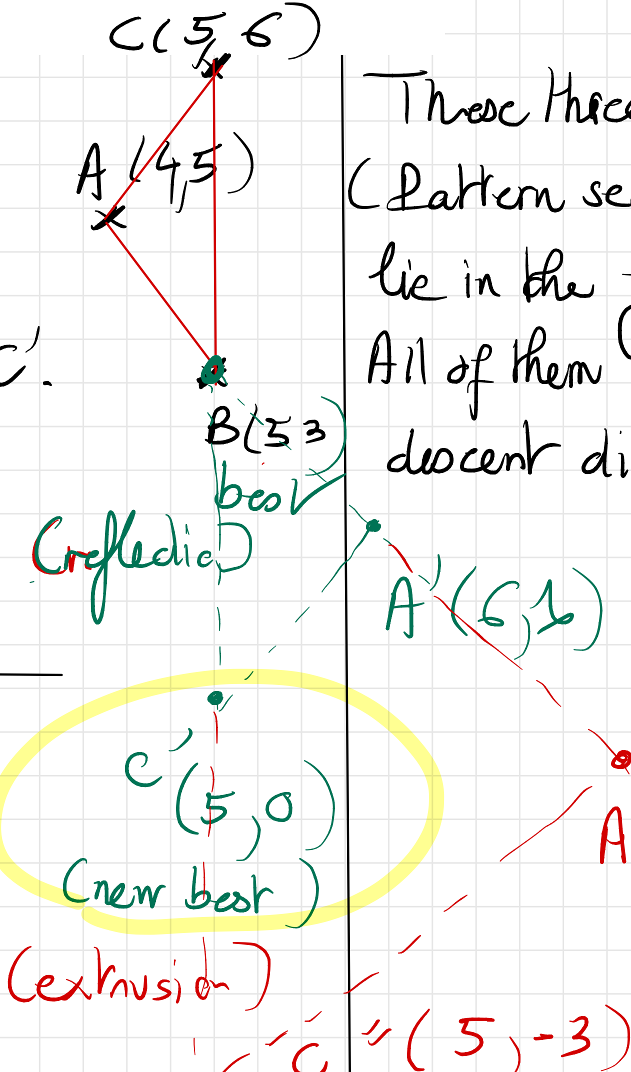
Case 1 Case 2 Case 3 Case 4

However, on a parallel computer, the MDS evaluation can be done in parallel.

Exercice: for the same example, make 1 iteration of MDS:

$\Longrightarrow$ A MDS iteration gives the new triangle $A'BC'$. (reflected)





These three methods (Pattern search, Nelder Mead, MDS) lie in the family of "direct methods". All of them try to find a correct descent direction.

In §3, another idea is investigated: with trust region method.

$\bigoplus$ Small quizz on direct search methods