Final Exam - Derivative Free Optimization (Part I)

Anne Auger
anne.auger@inria.fr

February 2017

The number of points is indicative. The answers should be carefully justified.

Exercice 1 (7 points)

We consider the following test functions:

- $f_1(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{n} x_i^2$
- $f_2(\mathbf{x}) = \left(\frac{1}{2}\sum_{i=1}^{n} x_i^2\right)^4$
- $f_3(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{n} 10000^{\frac{i-1}{n-1}} x_i^2$

1. Give for f_1 and f_3 the Hessian matrix and its condition number.

   In order to minimize the functions f_1, f_2, f_3 in dimension $n = 10$, we are using the (1+1)-ES algorithm with one-fifth success rule for adapting the step-size (no covariance matrix adaptation mechanism is used, only step-size adaptation). The initial step-size σ_0 is set to 10 and the initial mean vector to $(100, 100, \ldots, 100)^T$. We are running the algorithm 5 times independently on each function and we report the number of calls to the function (or number of function evaluations) that the algorithm needs to reach a function value strictly smaller than 10^{-6}. The results are presented in the following table

function	# Evals to reach 10^{-6} for 5 different runs				
f_1	830	825	946	695	749
f_2	489	566	537	509	378
f_3	304480	223808	235580	194545	282329

2. Comment the differences observed between f_1, f_2 and f_3.

3. Why do we observe a difference between f_1 and f_2? How can we change the stopping criterion to not see a difference anymore?

4. Why do we observe a difference between f_1 and f_3? Which algorithm could improve the results observed on f_3? [explain].

We consider now the functions

- $f_4(\mathbf{x}) = 10^4 x_1^2 + \sum_{i=2}^n x_i^2$
- $f_5(\mathbf{x}) = f_4(\mathbf{R}\mathbf{x})$, where $\mathbf{R} \in \mathcal{M}_n(\mathbb{R})$ is a rotation matrix sampled randomly.

We are using the CMA-ES algorithm to minimize those two functions as well as a variant of CMA-ES called sep-CMA-ES where at each iteration the covariance matrix C for sampling candidate solutions is **diagonal**.

5. Give the geometric shape of the iso-density lines of the Gaussian vector used to sampled candidate solutions in the sep-CMA-ES algorithm.

   In dimension $n = 10$, we initialize both algorithms setting the mean vector to $(100, 100, \ldots 100)^T$, the initial step-size to 10 and the initial covariance matrix to the identity. We are running the algorithm three times independently. We report the number of function evaluations to reach a function value strictly smaller than 10^{-6}. The results are presented in the following table:

	# Evals to reach 10^{-6} for 3 different runs					
function	CMA-ES			sep-CMA-ES		
f_4	4242	3902	4322	2172	2082	2512
f_5	4062	4262	4002	161072	168222	157132

6. Comment for both algorithms the differences observed between f_4 and f_5.
7. Why do we observe such a big difference between f_4 and f_5 for the sep-CMA-ES algorithm. Why don't we observe such a difference for the CMA-ES algorithm?
8. How can we explain that the sep-CMA-ES algorithm is faster than the CMA-ES algorithm on the function f_4?

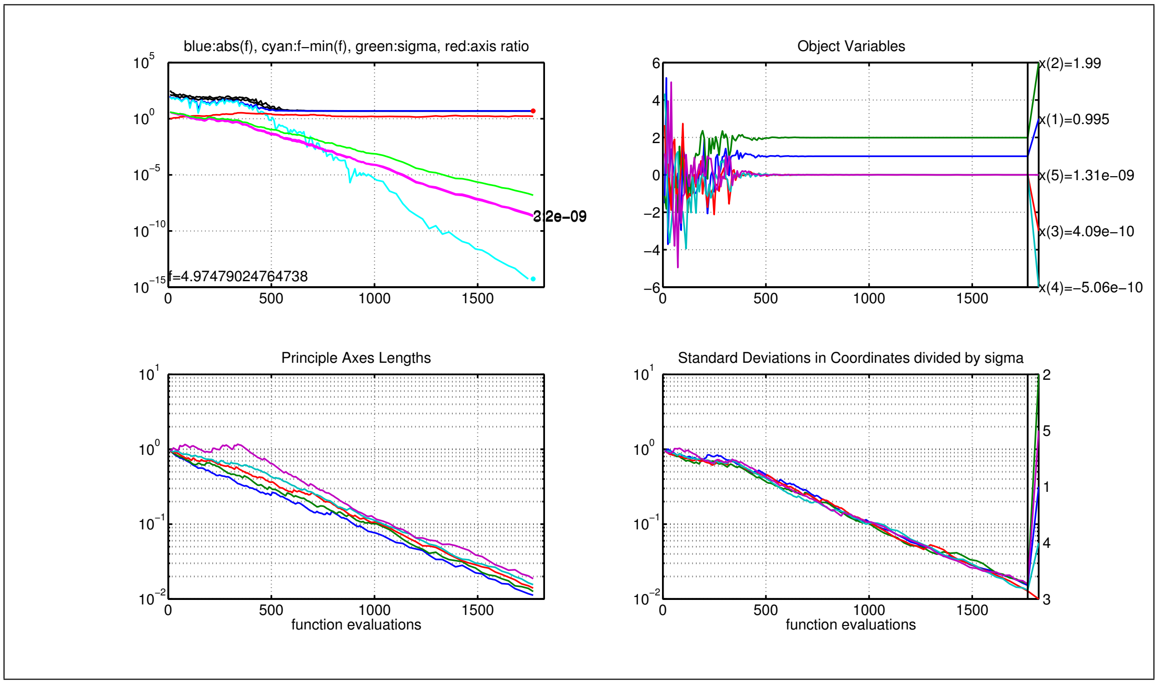
Exercice 2 (3 points)

We consider the Rastrigin test function defined as

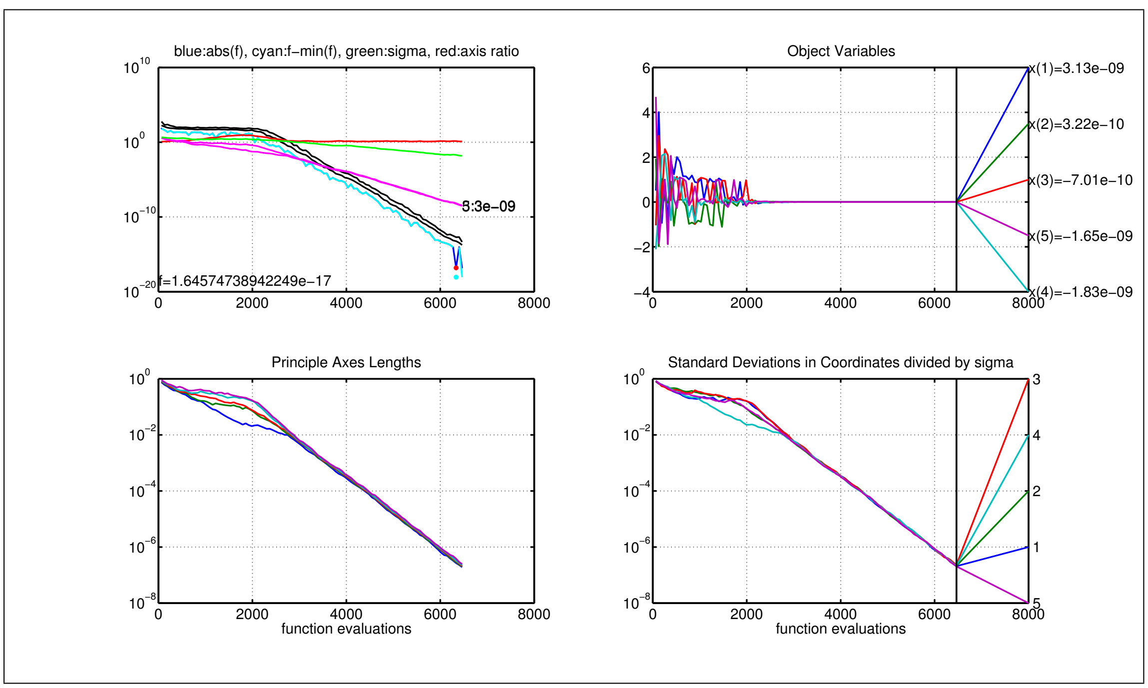
$$f(\mathbf{x}) = 10n + \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi \mathbf{x}_i))$$

1. What is the optimum of f?
2. Is the function separable, multimodal? [We expect a small proof to justify the answers]

   The CMA-ES algorithm is used to minimize the Rastrigin function in dimension $n = 5$. It is initialized with a mean vector equal to $(1, \ldots, 1)$ and a step-size equal to 5. Two trials are performed, the first one using the default population size of CMA-ES, that is $\lambda = 8$ in dimension 5. The second one with a larger population size equal to $\lambda = 64$. The trials are presented in Figure 1 (one trial on top, one trial below).
3. Are both trials converging to the global optimum of the function? [explain]
4. Identify which figure correspond to the trial with population size equal to $\lambda = 8$ and which figure correspond to the trial with population size equal to $\lambda = 64$. [explain your reasonning]



A



B

Figure 1: Single trials of the CMA-ES algorithm on the Rastrigin function. Identify the population size used for each trial.

DERVATIVE FREE OPTIMIZATION FINAL EXAM, PART 2

Exercice 1 *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 :

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

Exercice 2 *On the Lagrange interpolation*

Consider a set $\mathcal{Y} = \{X_1, ..., X_p\}$ of p points in $\mathbb{R}^n$ where p is the cardinality of the polynomial space $\mathbb{R}_d[x_1, ..., x_n]$ $(d \geq 1)$. Assume that the set is poised. Denote $\mathcal{B} = \{\Phi_1, ..., \Phi_p\}$ the monomial basis of $\mathbb{R}_d[x_1, ..., x_n]$.

The following algorithm is proposed to define a new polynomial basis :

Initialisation : set $l_j = \Phi_j$ for all $j = 1, ..., p$.

For $i = 1, 2, ..., p$:

- *Point selection* : find $j_0 = argmax_{i \leq j \leq p} |l_i(X_j)|$. If $l_i(X_{j_0}) = 0$ then stop (the set is not poised). Otherwise, swap points X_i and X_{j_0} in $\mathcal{Y}$.
- *Normalisation* : change $l_i(x) \leftarrow \dfrac{l_i(x)}{l_i(X_i)}$
- *Orthogonalization* : for $j = 1, ..., p$, $j \neq i$, change $l_j(x) \leftarrow l_j(x) - l_j(X_i) l_i(x)$

1. If $d \in \{1, 2\}$, what is the value of p for a given n ?
2. Give a condition on a matrix, built with $\mathcal{B}$ and $\mathcal{Y}$, so that the set is poised.
3. Prove that the previous algorithm transforms the basis $\mathcal{B}$ into the Lagrange basis (which definition will be recalled).

Exercice 3 *On a first order DFO trust region method*

The following algorithm in Matlab gives an example of a first order DFO trust region method. The objective is here to use a classical trust region method in dimension n, based on a linear interpolation of the function to minimize f made with a Lagrange interpolation from a set of p points :

```
n=3;  % dimension
p=n+1;
gamma=1.1;
theta=0.9;
eta=0.01;
Nstep=100;
X=rand(n,1); delta=0.1;           % initialization
Xla=[X,X*ones(1,p-1)+delta*(ones(n,p-1)-2*rand(n,p-1))];
Xlatot=Xla;  % total set of possible interpolation points
Xtot=[X];
for i=1:Nstep
    k=size(Xlatot,2);
    u=zeros(k,1);
    for j=1:k
        u(j)=norm(Xlatot(:,j)-X);
    end
    [a,b]=sort(u);
    Xla=Xlatot(:,b(1:p));      % choice of the nearest p points from X
    w=linlagrange(X,Xla);g=w(2:p);A=zeros(n,n);b=zeros(n,1);
    hplus=linprog(g,A,b,A,b,-delta*ones(n,1),delta*ones(n,1));
    Xplus=X+hplus;
    Xlatot=[Xlatot,Xplus];
    rhok=(f(X)-f(Xplus))/(f(X)-linmodel(g,f(X),hplus)+1E-16);
    if (rhok>eta)
       X=Xplus;delta=gamma*delta;
    else
       delta=theta*delta;
    end
    Xtot=[Xtot,X];
end
disp('best value:');disp(X)
```

In particular, the Matlab instruction `linprog` is used to minimize the function $m(x) = g' * x$ for $-\delta \leq x_i \leq \delta$. ($1 \leq i \leq n$). The functions `linlagrange`, `linmodel` and `f` need to be defined to complete the code.

1. Give a global description of the script above.
2. Write a possible function `linmodel.m`
3. Write a possible function `linlagrange.m`, either in the particular case where $n = 2$ or in the general case.