



**Nelder-Mead
User's Manual
– The Fminsearch Function –**

Michaël BAUDIN

Version 0.3
September 2009

Contents

1	The <i>fminsearch</i> function	3
1.1	Some notes about <i>fminsearch</i> 's algorithm	3
1.1.1	The algorithm	3
1.1.2	The initial simplex	3
1.1.3	The number of iterations	3
1.1.4	The termination criteria	4
1.2	Numerical experiments	4
	Bibliography	6

Chapter 1

The *fminsearch* function

1.1 Some notes about *fminsearch*'s algorithm

1.1.1 The algorithm

The algorithm used is the Nelder-Mead algorithm. This corresponds to the "variable" value of the "method" option of the *neldermead*.

1.1.2 The initial simplex

The *fminsearch* algorithm uses a special initial simplex, which is an heuristic depending on the initial guess. The strategy chosen by *fminsearch* corresponds to the `-simplex0method` flag of the *neldermead* component, with the "pfeffer" method. It is associated with the `-simplex0deltausual = 0.05` and `-simplex0deltazero = 0.0075` parameters. Pfeffer's method is an heuristic which is presented in "Global Optimization Of Lennard-Jones Atomic Clusters" by Ellen Fan [1]. It is due to L. Pfeffer at Stanford. See in the help of *optimsimplex* for more details.

1.1.3 The number of iterations

In this section, we present the default values for the number of iterations in *fminsearch*.

The `options` input argument is an optionnal data structure which can contain the `options.MaxIter` field. It stores the maximum number of iterations. The default value is $200n$, where n is the number of variables. The factor 200 has not been chosen by chance, but is the result of experiments performed against quadratic functions with increasing space dimension.

This result is presented in "Effect of dimensionality on the nelder-mead simplex method" by Lixing Han and Michael Neumann [3]. This paper is based on Lixing Han's PhD, "Algorithms in Unconstrained Optimization" [2]. The study is based on numerical experiment with a quadratic function where the number of terms depends on the dimension of the space (i.e. the number of variables). Their study shows that the number of iterations required to reach the tolerance criteria is roughly $100n$. Most iterations are based on inside contractions. Since each step of the Nelder-Mead algorithm only require one or two function evaluations, the number of required function evaluations in this experiment is also roughly $100n$.

1.1.4 The termination criteria

The algorithm used by *fminsearch* uses a particular termination criteria, based both on the absolute size of the simplex and the difference of the function values in the simplex. This termination criteria corresponds to the "tolssizedeltafvmethod" termination criteria of the *neldermead* component.

The size of the simplex is computed with the $\sigma - +$ method, which corresponds to the "sigmaplus" method of the *optimsimplex* component. The tolerance associated with this criteria is given by the "TolX" parameter of the *options* data structure. Its default value is 1.e-4.

The function value difference is the difference between the highest and the lowest function value in the simplex. The tolerance associated with this criteria is given by the "TolFun" parameter of the *options* data structure. Its default value is 1.e-4.

1.2 Numerical experiments

In this section, we analyse the behaviour of Scilab's *fminsearch* function, by comparison of Matlab's *fminsearch*. We especially analyse the results of the optimization, so that we can check that the algorithm is indeed behaving the same way, even if the implementation is completely different. Our test case is based on Rosenbrock's function.

The following Matlab script allows to see the behaviour of Matlab's *fminsearch* function on Rosenbrock's test case.

```
format long
banana = @(x)100*(x(2)-x(1)^2)^2+(1-x(1))^2;
[x,fval,exitflag,output] = fminsearch(banana,[-1.2, 1])
output.message
```

When this script is launched in Matlab, the following output is produced.

```
>> format long
>> banana = @(x)100*(x(2)-x(1)^2)^2+(1-x(1))^2;
>> [x,fval] = fminsearch(banana,[-1.2, 1])
>> [x,fval,exitflag,output] = fminsearch(banana,[-1.2, 1])
x =
    1.000022021783570    1.000042219751772
fval =
    8.177661197416674e-10
exitflag =
    1
output =
    iterations: 85
    funcCount: 159
    algorithm: 'Nelder-Mead simplex direct search'
    message: [1x194 char]
>> output.message
ans =
Optimization terminated:
the current x satisfies the termination criteria using
OPTIONS.TolX of 1.000000e-04
and F(X) satisfies the convergence criteria using
OPTIONS.TolFun of 1.000000e-04
```

The following Scilab script allows to solve the problem with Scilab's *fminsearch*.

```
format(25)
function y = banana (x)
    y = 100*(x(2)-x(1)^2)^2 + (1-x(1))^2;
endfunction
[x , fval , exitflag , output] = fminsearch ( banana , [-1.2 1] )
output.message
```

The output associated with this Scilab script is the following.

```
-->format(25)
-->function y = banana (x)
--> y = 100*(x(2)-x(1)^2)^2 + (1-x(1))^2;
-->endfunction
-->[x , fval , exitflag , output] = fminsearch ( banana , [-1.2 1] )
output =
    algorithm: "Nelder-Mead_simplex_direct_search"
    funcCount: 159
```

```

    iterations: 85
    message: [3x1 string]
    exitflag =
    1.
    fval =
    0.00000000008177661099387
    x =
    1.0000220217835567027009    1.0000422197517710998227
-->output.message
    ans =

!Optimization terminated:
!
!the current x satisfies the termination criteria using OPTIONS.TolX of 1.000000e-004
!
!and F(X) satisfies the convergence criteria using OPTIONS.TolFun of 1.000000e-004
!
```

The result is as close as possible to the result produced by Matlab. More specifically :

- the optimum x is the same up to 14 significant digits,
- the function value at optimum is the same up to 9 significant digits,
- the number of iterations is the same,
- the number of function evaluations is the same,
- the exit flag is the same,
- the content of the output is the same (but the string is not display the same way).

Bibliography

- [1] Ellen Fan. Global optimization of lennard-jones atomic clusters. Technical report, McMaster University, February 2002.
- [2] Lixing Han. *Algorithms in Unconstrained Optimization*. Ph.D., The University of Connecticut, 2000.
- [3] Lixing Han and Michael Neumann. Effect of dimensionality on the nelder-mead simplex method. *Optimization Methods and Software*, 21(1):1–16, 2006.