# Evolutionary Algorithms
# Problem Set - Evolution Strategies in MATLAB

Below you find the skeleton of a (1+1)-ES implementation in MATLAB:

```
function [xp, fp, stat] = es(fitnessfct, n, lb, ub, stopeval)

  % Strategy parameters
  ...

  % Initialize
  xp = ...
  fp = ...
  sigma = ...
  evalcount = 0;

  % Statistics administration
  stat.name = '(1+1)-ES';
  stat.evalcount = 0;
  stat.histsigma = zeros(1, stopeval);
  stat.histf = zeros(1, stopeval);

  % Evolution cycle
  while evalcount < stopeval

    % Generate offspring and evaluate
    xo = ... % generate offspring from parent xp
    fo = ... % evaluate xo using fitnessfct
    evalcount = evalcount + 1;

    % select best and update success-rate and update stepsize
    % Important: MINIMIZATION!

    % Statistics administration
    stat.histsigma = % stepsize history
    stat.histf = % fitness history

    % if desired: plot the statistics

  end

end
```

1. Complete the code with the update of the stepsize ($\sigma$) according to the 1/5th success rule and run the (1+1)-ES on the Sphere function (minimization) using $n = 10$, stopeval $= 10000$, lb $= [-10]^n$, ub $= [10]^n$:

   ```
   function f = sphere(x)
     f = sum(x.^2);
   end
   ```

2. Plot fitness against evaluations for one run of the ES. Is there a point in time at which the algorithm does not converge anymore?

3. Plot stepsize against evaluations for one run of the ES. How does it develop over time and what would you conclude from this?

4. Below you find two scripts. The script `multiple_runs` runs the (1+1)-ES multiple times and stores the statistics of those runs in the file `statistics.mat`. The script `plot_statistics` generates a plot of the mean fitness vs. evaluations.

```
function [] = multiple_runs(fitnessfct, n, lb, ub, stopeval, runs)
  for i = 1 : runs
    [xopt, fopt, stat(i)] = es(fitnessfct, n, lb, ub, stopeval);
  end
  save('statistics.mat', stat)
end

function [] = plot_statistics(stat, fitnessfct, n, lb, ub, stopeval, runs)
  for i = 1 : runs
    histf(:,i) = stat(i).histf(1:stopeval);
  end
  plot(mean(histf, 2))
  xlabel('evaluations')
  ylabel('fitness')
end
```

Use these scripts to obtain a mean fitness plot of 10 runs on the Sphere function (again with N = 10, stopeval = 10000, lb = $[-10]^N$, ub = $[10]^N$).

5. Run the scripts for the Ackley function and the Rosenbrock function:

```
function f = ackley(x)
  [ps,n] = size(x);
  f = 20 - 20 * exp(-0.2 * sqrt(sum(x .^ 2, 2) / n)) ...
        - exp(sum(cos(2 * pi * x), 2) / n) + exp(1);
end

function f = rosenbrock(x)
  f = 100 * sum((x(1:end-1).^2 - x(2:end)).^2) + sum((x(1:end-1)-1).^2);
end
```

6. Create a $(1, \lambda)$-ES implementation and compare it with the (1+1)-ES on the Sphere function (hint: plot mean fitness vs. evaluations for both algorithms in one figure).

7. Compare both ESs on the Ackley and Rosenbrock functions.