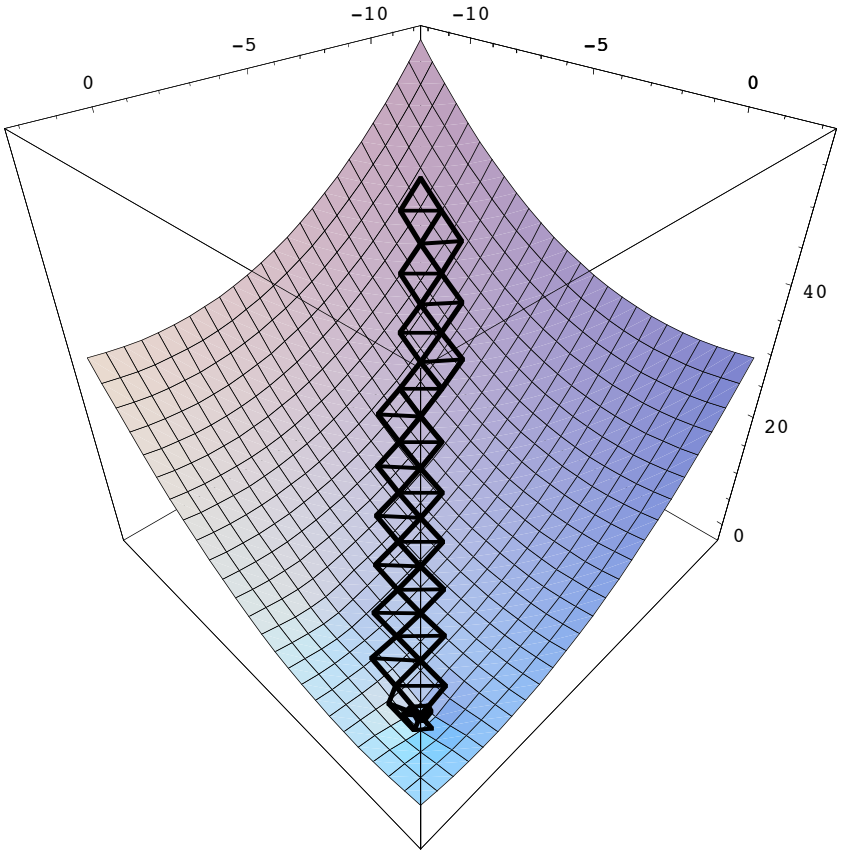
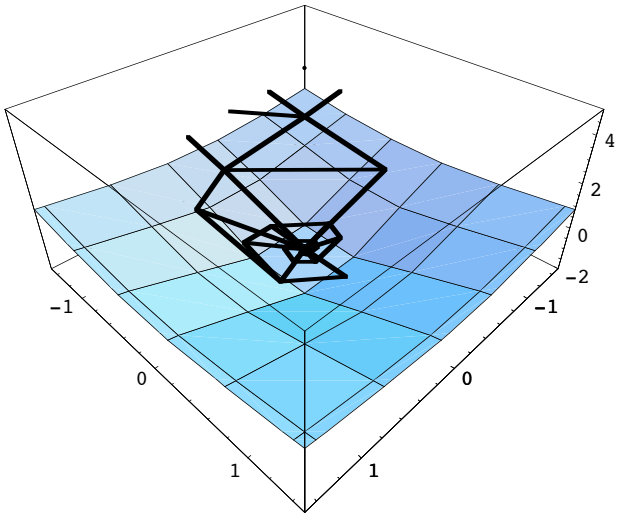
Nelder-Mead

```
f[x_, y_] := 0.2 (x^2 + y^2) - 2 Exp[-(x^2 + y^2)] - Exp[-(x^2 + (y + 4)^2) / 2];

vertexOperate[simplex_, vertexNum_, op_] := Insert[#2, #1, vertexNum] & @@
   op[Part[simplex, vertexNum], Delete[simplex, vertexNum]];

move[param_][this_, others_] := {this + (1 + param) * (Mean[others] - this), others};

shrink[this_, others_] := {this, (# + this) / 2 & /@ others};

extremeVertex[simplex_, f_, p_] := Ordering[f @@@ simplex, 1, p][[1]];

simplexCompare[v_, f_, p_][a_, b_] := (f @@ a[[v]]) ~p~ (f @@ b[[v]]);

not[p_][x_, y_] := ! p[x, y];

improve[a_, b_, p_] := If[b ~p~ a, a = b; True, False];
SetAttributes[improve, HoldFirst];

simplexGraphics[simplex_] := Line[Append[simplex, First[simplex]]];
simplexGraphics3D[simplex_, f_] :=
  Line[Append[#, f @@ #] & /@ Append[simplex, First[simplex]]];

updateSimplex[simplex_, f_, p_] :=
  Module[{v = extremeVertex[simplex, f, not[p]], curSimplex = simplex, comparator},
   comparator = simplexCompare[v, f, p];
   If[improve[curSimplex, vertexOperate[curSimplex, v, move[1]], comparator],
    improve[curSimplex, vertexOperate[curSimplex, v, move[2]], comparator]; curSimplex,
    If[improve[curSimplex, vertexOperate[curSimplex, v, move[1 / 2]], comparator],
     curSimplex,
     If[improve[curSimplex, vertexOperate[curSimplex, v, move[-1 / 2]], comparator],
      curSimplex,
      vertexOperate[curSimplex, extremeVertex[simplex, f, p], shrink]]]]];

path =
  NestList[updateSimplex[#, f, Less] &, {{-10., -10.}, {-9., -10.}, {-10., -9.}}, 200];
```

```
plot = Show[Graphics3D[
    {AbsoluteThickness[2.], Map[# + {0, 0, 1} &, simplexGraphics3D[#, f], {2}] & /@ path}],
   Block[{$DisplayFunction = Identity}, Plot3D[f[x, y], {x, -12, 2}, {y, -12, 2}]],
   Axes → True, BoxRatios → {1, 1, 1}, ViewPoint → {1, 1, 1}];
```

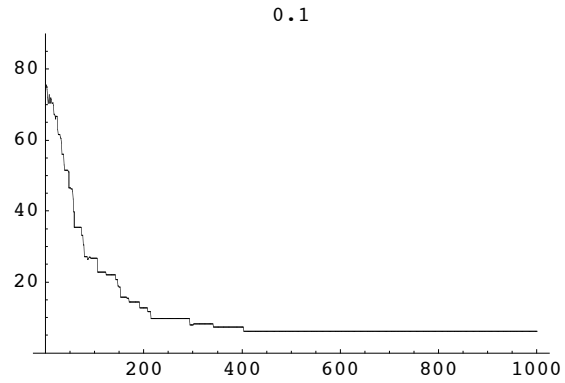


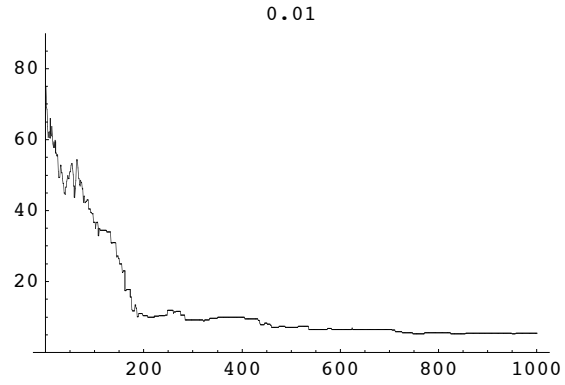

```
Show[plot, PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}, {-2, 5}}, BoxRatios → {2, 2, 1}];
```

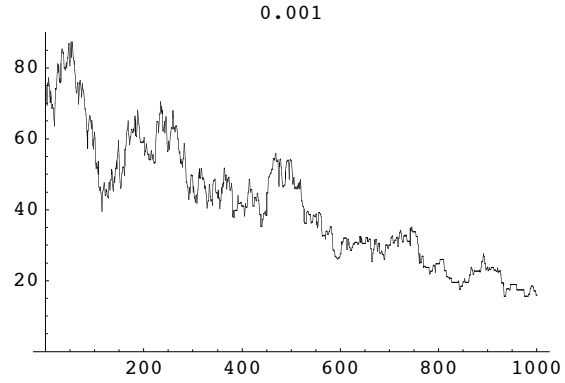


Spin Glass – Annealing

```
Needs["Statistics`ContinuousDistributions`"];

J = RandomArray[NormalDistribution[0, 1], 100];
minEnergy = -Total[Abs[J]];
glass = Table[(-1)^Random[Integer, {1, 2}], {100}];

glassEnergy[glass_, J_] := -J.(Times @@@ Partition[Append[glass, First[glass]], 2, 1]);
flipRandom[glass_] := MapAt[-# &, glass, Random[Integer, {1, Length[glass]}]];
updateGlass[oldGlass_, beta_, J_] :=
  (Function[{newGlass, oldEnergy, newEnergy}, If[newEnergy < oldEnergy, newGlass,
        If[Random[] < Exp[-beta (newEnergy - oldEnergy)], newGlass, oldGlass]]][#,
      glassEnergy[oldGlass, J], glassEnergy[#, J]]) &[flipRandom[oldGlass]];
runGlass[glass_, iters_, alpha_, J_] := First /@ NestList[
    {updateGlass[#[[1]], alpha * #[[2]], J], #[[2]] + 1} &, {glass, 0}, iters];

plotRange = {0, glassEnergy[glass, J] - minEnergy} * 1.2;
alphas = {0.1, 0.01, 0.001};
Table[ListPlot[(glassEnergy[#, J] & /@ runGlass[glass, 1000, alphas[[i]], J]) - minEnergy,
   PlotJoined → True, PlotRange → plotRange,
   PlotLabel → ToString[alphas[[i]]]], {i, Length[alphas]}];
```

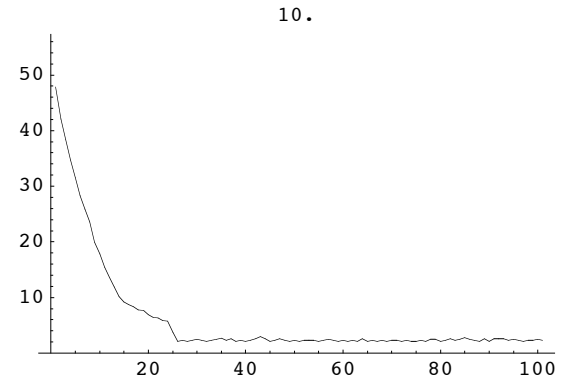


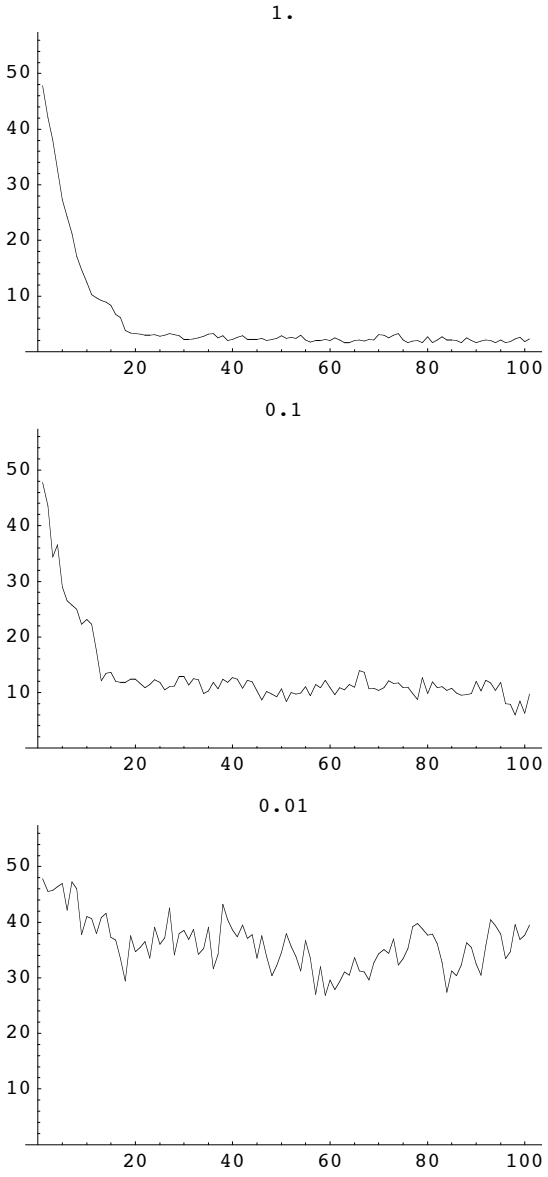




Spin Glass – Genetic Algorithm

```
breed[glass1_, glass2_] := Function[{crossover},
    flipRandom[Join[Take[glass1, crossover], Drop[glass2, crossover]]]][
   Random[Integer, {0, Length[glass1]}]];
pickRandom[choices_, probs_] := Module[
   {cum = Rest[FoldList[Plus, 0, probs]], rand = Random[Real, {0, Total[probs]}]},
   choices[[Position[cum, _?(# > rand &), {1}, 1][[1, 1]]]];
updatePopulation[pop_, J_, beta_] := Module[{minEnergy = -Total[Abs[J]], probs},
   probs = Exp[-beta (glassEnergy[#, J] - minEnergy)] & /@ pop;
   Table[breed[pickRandom[pop, probs], pickRandom[pop, probs]], {Length[pop]}]];
runGenGlass[population_, iters_, beta_, J_] :=
  NestList[updatePopulation[#, J, beta] &, population, iters];
minPopEnergy[pop_, J_] := Min[glassEnergy[#, J] & /@ pop];

population = Table[(-1)^Random[Integer, {1, 2}], {100}, {100}];

plotRange = {0, minPopEnergy[population, J] - minEnergy} * 1.2;
betas = {10., 1., 0.1, 0.01};
Table[
  ListPlot[(minPopEnergy[#, J] & /@ runGenGlass[population, 100, betas[[i]], J]) - minEnergy,
   PlotJoined → True, PlotRange → plotRange,
   PlotLabel → ToString[betas[[i]]], {i, Length[betas]}];
```




1.
50
40
30
20
10
20
40
60
80
100
0.1
50
40
30
20
10
20
40
60
80
100
0.01
50
40
30
20
10
20
40
60
80
100