SLang - the Next Generation

Tutorial

Christian Bucher, Sebastian Wolff
Center of Mechanics and Structural Dynamics
Vienna University of Technology

November 8, 2010

0.1 Optimization with constraints

As as simple example, consider an optimization problem as follows: Minimize

$$f(x_1, x_2) = (x_1 + 1)^2 + x_1^2 x_2^2 + \exp(x_1 - x_2) \tag{1}$$

subject to the constraint condition

$$-\frac{x_1^2}{2} - x_2 + 1.5 < 0 \tag{2}$$

The objective function and the feasible domain are shown in Fig. **??** The procedure to arrive at the solution of

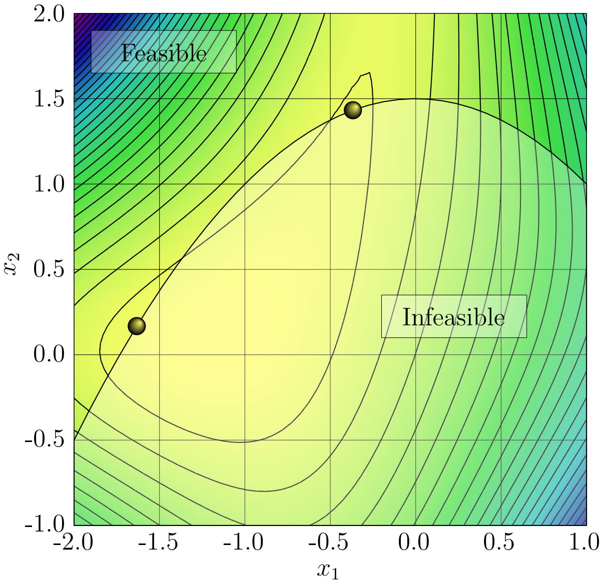

Figure 1: Objective function and feasible domain

this problem is given in the following script.

```
--[[
SLangTNG
Simple test example for optimization
(c) 2009 Christian Bucher, CMSD-VUT
--]]

-- This function defines the objective
  function objective (x)
    local a = (x[0]+1)^2+x[0]^2*x[1]^2+math.exp(x[0]-x[1])
    return a
    end

-- This function defines the constraints. Note that it returns an array
  function constraints (x)
    local a = tmath.Matrix(1)
    a[0] = -x[0]^2/2-x[1]+1.5
    return a
    end

-- Main program starts here
-- Create an optimization object an set the starting value
-- The optimization algorithm is CONMIN by G. Vanderplaats
  nvariables = 2; nconstraints = 1
  ops=optimize.Conmin(nvariables, nconstraints)
  start=tmath.Matrix({{-1},{0}});
  ops:SetDesign(start)

-- Run optimization in reverse communication mode
-- This is an endless loop which is terminated when
-- the value "go_on" returned from Compute is equal to zero
  go_on=1
```

```
  while(1) do
-- Compute one step and check for termination
    go_on=ops:Compute()
    if (go_on==0) then break end

-- Compute objective
    x = ops:GetDesign()
    obj=objective(x)
    ops:SetObjective(obj);

-- Compute constraints
    cons = constraints(x)
    ops:SetConstraints(cons)
    end

-- Print optimization result
  sol = ops:GetDesign()
  print("sol", sol)
```

Starting at the point $\mathbf{x} = [-1, 0]$ we get the solution $\mathbf{x}^* = [-1.633, 0.168]$. This happens to be the global minimum. Choosing different starting points (e.g. at the origin) may lead to a different solution (i.e. the second local mimimum).