Universität Konstanz
Fachbereich Mathematik und Statistik
Prof. Dr. Stefan Volkwein
Martin Gubisch, Roberta Mancini, Stefan Trenz

23rd May 2011

Optimization
Programming 2

Implement the Gradient Projection Algorithm.

The gradient projection algorithm is a modified version of the steepest descent algorithm in which only solutions that lie in a closed bounded domain Ω are valid. The main idea is to project what was the update in the steepest descent method on the domain Ω, i.e. $x_{k+1} = P(x_k + td_k)$, where t is the step length calculated using a step size strategy. As in the steepest descent method, we consider $d_k = -\nabla f(x_k)/\|\nabla f(x_k)\|$.

Defining as in Exercise 14 (b)

$$x(t) = P(x + td),$$

we use the following termination condition for the line search (see Exercise 15)

$$f(x(t)) - f(x) \leq -\frac{\alpha}{t}\|x - x(t)\|^2. \tag{1}$$

The pseudo-code of the gradient projection algorithm looks like

```
while the termination criteria are not fulfilled
  find stepsize t using the line search strategy
  set x = x(t)
end
```

The termination criteria are

```
||x-x(1)|| < epsilon       or
||grad(f)(x)|| < epsilon   or
number iterations > MAX number of iterations
```

Implement this method using the following steps.

Part 1: Generate a file `projection.m` and implement the function

```
function x = projection(x0, a, b)
```

with the current point `x0`, lower bound `a` and upper bound `b`. The function should return the projected point `x` according to the projection

$$P : \mathbb{R}^2 \to \Omega := \{x \in \mathbb{R}^2 \mid a_1 \leq x_1 \leq b_1,\ a_2 \leq x_2 \leq b_2\}$$

(cf. Exercise 14 (b) with $L = a$ and $R = b$).

Test your program by calculating $P(x_0)$ for a rectangular domain defined by the lower bound (lower left corner) `a=[-1;-1]`, the upper bound (upper right corner) `b=[1;1]`, a direction `d=[1.5;1.5]`, step sizes `t=0` and `t=1` and the following points $x_0 = y + t * d \in \mathbb{R}^2$:

Points y:	Projections for $t = 0$	Projections for $t = 1$
`[-2;-2]`	(-1,-1)	(-0.5,-0.5)
`[-1;-1]`	(-1,-1)	(0.5,0.5)
`[-0.5;0.5]`	(-0.5,0.5)	(1,1)
`[2;0.5]`	(1,0.5)	(1,1)
`[1;-0.5]`	(1,-0.5)	(1,1)

Table 1: Testing points and their projections with respect to `t`

Part 2: Implement the gradient projection algorithm with direction $d_k := -\frac{\nabla f(x_k)}{\|\nabla f(x_k)\|}$. Modify therefore the Armijo step size strategy according to the projection rule (1) and save it to a file `modarmijo.m` containing the function

```
function t = modarmijo(fhandle, x0, d, t0, alpha, beta, a, b)
```

with initial point `x0`, descent direction `d`, initial step size `t0`, `alpha` and `beta` for the Armijo rule and `a` and `b` for the projection rule.

Generate a file `gradproj.m` for the function

```
function X = gradproj(fhandle, x0, epsilon, N, t0, alpha, beta, a, b)
```

with initial point `x0`, parameter `epsilon` for the termination condition $\|\nabla f(x_k)\| < \epsilon$ and the additional termination condition $\|x - x(1)\| \leq \epsilon$, `N` for the maximal number of iteration steps, parameters `t0`, `alpha` and `beta` for the Armijo rule and `a` and `b` for the projection rule. Modify therefore the gradient method `gradmethod.m` from program 1.

The program should return a matrix `X = [x0; x1; x2; ...]` containing the whole iterations.

Test your program by using the Rosenbrock function from program 1 with the following parameters and write your observations in the written report:

`epsilon=1.0e-2`, `N=1.0e4`, `t0=1`, `alpha=1.0e-2`, `beta=0.5` and

a) `x0=[1;-0,5]`, `a=[-1;-1]` and `b=[2;2]`

b) `x0=[-1;-0,5]`, `a=[-2;-2]` and `b=[2;0]`

Deadline: Tuesday, 31st May, 10:00 am