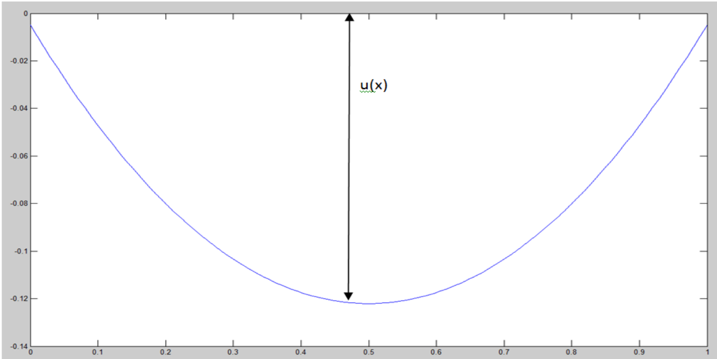
The obstacle Problem

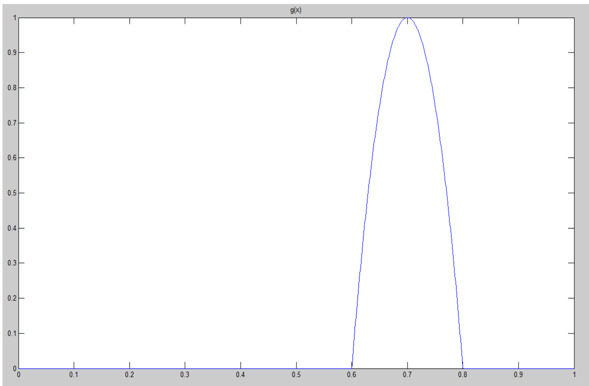
Let g be a continuous function in [0,1]. The obstacle problem will be:

$$\text{find u : } [0,1] \to \mathbb{R} \text{ such as } \begin{cases} -u''(x) \geq 1 & x \in [0,1] \\ u(x) \geq g(x) & x \in [0,1] \\ (-u''(x) - 1)(u(x) - g(x)) = 0 & x \in [0,1] \\ u(0) = u(1) = 0 & \end{cases}$$



Equation of a rope clamped at the extremities

This figure represents a rope subject to its weight (equal to 1 here) and clamped at the extremities $x = 0$ and $x = 1$. We add an obstacle in the problem and the rope has to be above this.



Representation of the obstacle

In this case $g(x) = max(0, 1 - 100(x - 0,7)^2)$.

Discretization

We discretize this problem by introducing an uniform mesh: $x_j = jh$ where h is the step of the mesh and $j \in \{0, ..., n+1\}$ where n is a integer. We set $h = \frac{1}{n+1}$ and g_j=g(x_j).

The problem becomes:

$$\text{find } u_j = u(x_j) \text{ such as } \begin{cases} -\frac{u_{j-1}-2u_j+u_{j+1}}{h^2} \geq 1 & j \in \{0, .., n+1\} \\ u(x) \geq g(x) & j \in \{0, .., n+1\} \\ (-\frac{u_{j-1}-2u_j+u_{j+1}}{h^2} - 1)(u(x) - g(x)) = 0 & j \in \{0, .., n+1\} \\ u_0 = u_{n+1} = 0 \end{cases}$$

We introduce the matrix A and the vectors b and g:

$$\text{A}=\frac{1}{h^2}\begin{pmatrix} 2 & -1 & 0 & \dots & \dots & \dots & 0 \\ -1 & 2 & -1 & \ddots & & & \vdots \\ 0 & -1 & 2 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 2 & -1 & 0 \\ \vdots & & & \ddots & -1 & 2 & -1 \\ 0 & \dots & \dots & \dots & 0 & -1 & 2 \end{pmatrix}, \text{ b}=\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \text{ et g}=\begin{pmatrix} g_1 \\ \vdots \\ g_n \end{pmatrix}$$

with u=$\begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix}$ we have

$$\text{u is a solution of the problem} \Leftrightarrow \begin{cases} \min\limits_{v \in K} \frac{1}{2} < Av, v > - < b, v > \\ K = \{v \in \mathbb{R}^n : v \geq g\} \end{cases}$$

Let J be the functional in $\mathbb{R}^n$ such as $J(v) = \frac{1}{2} < Av, v > - < b, v >$

To solve this problem, we use the projected gradient method. This algorithm is very close to a gradient method with fixed step size. In each iteration the gradient method can gives us a X_k outside K. So we have to project the result with $\Pi_K(v)$. In this problem we used $\Pi_K(v) = max(v_i, g_i)$.

Algorithm 6 Projected gradient method

Require: $K_{max} \in \mathbb{N}$, $\mathbf{X}_0 \in \mathbb{R}^n$, [a,b], λ_1 and λ_n the eigenvalues

n=size(X_0)
i=0;
while $x_{k+1} - x_k > 10^{-4}$ **and** $k < K_{max}$ **do**
    for j=1:n **do**
        $X_j = max(X_j, g_j)$
    end for
    $\mathbf{x}_{k+1}=\mathbf{x}_k - \frac{2}{\lambda_1+\lambda_n}\nabla J_n(\mathbf{x}_k)$;
end while
return $\mathbf{x}_k$

λ_1 and λ_n are the first and last eigenvalues of the matrix A.

1. Test this algorithm for n = 5 to n = 100, display the number of iteration and the computing time regarding to n.
2. Perform some other tests when changing the obstacle.

Distance from a point to a line

We want to numerically determine the lowest distance from a point, $\mathbf{x}_0$ in dimension n to a hyperplane, $(\mathscr{H})$ defined by the equation $A\mathbf{x} = b$, where $A \in \mathscr{M}_{m,n}(\mathbb{R})$ with $m < n$.

The constraint minimization is written :

$$\min_{F(x)=0} J(\mathbf{x}).$$

With: $F(\mathbf{x}) = A\mathbf{x} + b$ and $J(\mathbf{x}) = \frac{1}{2}.^t(\mathbf{x} - \mathbf{x}_0).(\mathbf{x} - \mathbf{x}_0)$

The Lagrangian of this problem is :

$$L(\mathbf{x}) = J(\mathbf{x}) + \lambda.F(\mathbf{x})$$

At the optimum, the derivative of L is zero, so :

$$\nabla J(\mathbf{x}) + \lambda^* \nabla F(\mathbf{x}) = 0$$

$$\mathbf{x} - \mathbf{x}_0 + \lambda^* A^t = 0$$

$$A\mathbf{x} - \mathrm{x}_0 + \lambda^* AA^t = 0$$

$$b - \mathbf{x}_0 + \lambda^* AA^t = 0$$

$$\lambda^* AA^t = b - \mathbf{x}_0$$

$$\lambda^* = (AA^t)^{-1} b - \mathbf{x}_0$$

We replace λ^* by the expression obtained in the previous compute and we get:

$$\nabla J(\mathbf{x}^*) + \lambda^* \nabla F(\mathbf{x}) = 0$$

$$\mathbf{x}^* = \mathbf{x}_0 + A^t(AA^t)^{-1}(b - \mathbf{x}_0).$$

To compute the distance from a point to a line, we calculate $\mathbf{x}^* - \mathrm{x}_0$ and we take A like a row vector.

Then $AA^t = ||A||_2^2$ and $(AA^t)^{-1} = \frac{1}{||A||_2^2}$. By the same way, we have $A^t = ||A||_2$.

So $A^t(AA^t)^{-1}$ corresponds to $\frac{1}{||A||_2}$.

Finally, we have:

$$d(\mathbf{x}_0, \mathscr{H}) = \frac{||b - A\mathbf{x}_0||}{||A||}$$

The Uzawa algorithm is a kind of gradient method applied to the Lagrangian. Thereafter, a description of the algorithm where f et g are the constraints.

Algorithm 7 Uzawa algorithm

Require: $\rho > 0$, $K_{max} \in \mathbb{N}$ $\mathbf{x}_0 \in \mathbb{R}^n$

k=0;
while $|\mathbf{x}_{k+1} - \mathbf{x}_k| > 10^{-4}$ **and** $k < K_{max}$ **do**
  Compute $\mathbf{x}_k$ solution of min $L(\mathbf{x}, \mu_k, \lambda_k)$
  Compute μ_{k+1}, λ_{k+1} with:
  $\mu_{i,k+1} \leftarrow \mu_{i,k} - \rho f_i(\mathbf{x}_k)$; for $i = 1..p$
  $\lambda_{i,k+1} = \max(0, \lambda_{i,k} + \rho g_i(\mathbf{x}_k)$; for $j = 1..m$
  $k + 1 \leftarrow k$;
end while
return $\mathbf{x}_k$

1. test this algorithm for A=(1 1), b=2 and x0 =(0 0)';
2. Perform some other tests.
3. Use this algorithm to solve the obstacle problem.