Numerical Analysis – Computational Session

The Penalty Method for the Hanging Chain Problem

February 5, 2026

In this computational session, you will determine the equilibrium shape of a hanging chain by formulating and solving a constrained optimization problem. Starting from a physical model, you will derive the chain's potential energy and the associated geometric constraints, then reformulate the problem using a penalty method to obtain a sequence of unconstrained minimization problems. By implementing a gradient-based algorithm and varying the penalty parameter, you will explore how numerical solutions approach the constrained optimum and gain practical insight into the behavior of penalty methods in constrained optimization.

1 Problem Setup

A chain is suspended from two thin hooks that are 16 feet apart on a horizontal line, as shown in Figure 1. The chain itself consists of 20 links of stiff steel. Each link is 1 foot long (measured inside). We wish to formulate the problem to determine the equilibrium shape of the chain.

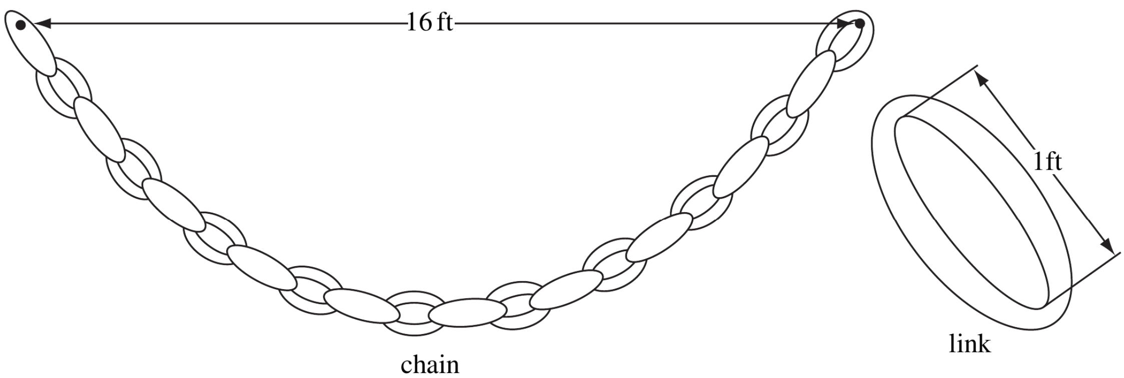

Figure 1: A hanging chain. From [1, Fig. 11.3].

The solution can be found by minimizing the chain's potential energy. Let us number the links consecutively from 1 to 20, starting with the left end. We let link i span an x distance of x_i and a y distance of y_i. Since each link has a fixed length of 1, the x_i can be expressed as a function of y_i. The potential energy of a link is its weight times its vertical height. The potential energy of the chain is the sum of the potential energies of each link. We choose the height of the left endpoint of the chain as reference, and assume that the mass of each link is concentrated at its center.

<u>**Your tasks**</u>**:**

1. Express the x_i as a function of y_i.
2. Assuming unit weight of each link, find the expression of the potential energy of the chain, denoted $U(\boldsymbol{y})$, with $\boldsymbol{y} := (y_1, y_2, \ldots, y_{19}, y_{20})$.
3. The chain is subject to the following two constraints:

   (a) The total y displacement is zero;

   (b) The total x displacement is 16.

   Using the notation above, express the constraints in terms of the variables y_i. Then, write down the constrained optimization problem, i.e., fill in the missing parts below:

$$\begin{aligned} \text{minimize} \quad & \ldots, \\ \text{subject to} \quad & \ldots, \\ & \ldots. \end{aligned}$$

   The equilibrium shape of the chain will be the solution of this constrained minimization problem.

2 The Penalty Method

To solve this constrained optimization problem using the penalty method, we convert it into a sequence of unconstrained minimization problems by adding penalty terms that strongly discourage violation of the constraints.

2.1 Penalty formulation

<u>**Your tasks**</u>**:**

1. Introduce a penalty parameter $\mu > 0$ and define the penalized objective function $\Phi_\mu(\boldsymbol{y})$.

   An unconstrained problem thus replaces the original constrained problem; write down the new formulation.
2. What happens when $\mu \to \infty$?

2.2 Algorithmic procedure

A typical penalty-method algorithm proceeds as follows:

1. **Initialization.** Choose an initial guess $\boldsymbol{y}^{(0)}$ and an initial penalty parameter μ_0.
2. **Unconstrained minimization.** For fixed μ_k, minimize $\Phi_{\mu_k}(\boldsymbol{y})$ using the gradient descent method.

3. **Penalty update.** Increase the penalty parameter, for example

$$\mu_{k+1} = c\,\mu_k, \qquad c > 1.$$

4. **Iteration.** Repeat until the constraint violations satisfy a certain tolerance, specified by a small constant $\varepsilon > 0$.

The penalty method algorithm is also described by the pseudocode in Algorithm 1.

Algorithm 1: Penalty Method for the Hanging Chain Problem

Input: Initial guess $\boldsymbol{y}^{(0)} \in \mathbb{R}^{20}$, initial penalty parameter $\mu_0 > 0$, penalty growth factor $c > 1$, tolerance $\varepsilon > 0$.

Output: Approximate equilibrium configuration $\boldsymbol{y}^*$.

$k \leftarrow 0$;

$\boldsymbol{y}^{(k)} \leftarrow \boldsymbol{y}^{(0)}$;

while *true* **do**

  Define the penalized objective function $\Phi_{\mu_k}(\boldsymbol{y})$;

  Compute

$$\boldsymbol{y}^{(k+1)} = \arg\min_{\boldsymbol{y}} \; \Phi_{\mu_k}(\boldsymbol{y})$$

  using the gradient descent method;

  if *1st constraint violation* $< \varepsilon$ **and** *2nd constraint violation* $< \varepsilon$ **then**

    return $\boldsymbol{y}^{(k+1)}$;

  end if

  $\mu_{k+1} \leftarrow c\,\mu_k$;

  $k \leftarrow k + 1$;

end while

Your tasks:

1. Implement the penalty-method algorithm to solve the unconstrained minimization problem. Follow the description above and the pseudocode in Algorithm 1. You may use a fixed step size or a backtracking line search, and stop the inner minimization when the norm of the gradient falls below a prescribed tolerance.

2. Use several values of the penalty parameter μ_k. To this aim, choose appropriate values for the initial penalty parameter μ_0 and the penalty growth factor c.

3. Plot the approximate equilibrium configurations $\boldsymbol{y}^*_{\mu_k}$ obtained for each value of μ_k that you tested. What do you observe?

References

[1] David G. Luenberger and Yinyu Ye. *Linear and Nonlinear Programming.* International Series in Operations Research & Management Science. Springer New York, NY, 3 edition, 2008.