The Brachistrochrone Problem: a Finite Element Approach

Harry Luo

March 29, 2025

Abstract

The Brachistochrone problem seeks the path $y(x)$ between two points that allows a particle sliding under gravity to travel in the minimum possible time. We present a numerical solution using the Finite Element Analysis (FEA) method. The continuous problem, formulated as minimizing the time functional

$$T[y(x)] = \int_0^{x_f} \frac{\sqrt{1 + (y'(x))^2}}{\sqrt{2gy(x)}} dx,$$

is discretized using P2 quadratic elements. This transforms the variational problem into a finite-dimensional nonlinear optimization problem, which is then solved using the L-BFGS-B algorithm.

1 Finite Element Discretization

We employ quadratic (P2) finite elements to discretize the time functional and approximate the unknown path $y(x)$.

First, we divide the domain $[0, x_f]$ into N uniform finite elements, each of length $h = x_f/N$. We define a total of $2N + 1$ global nodes along the domain. The coordinate of the m-th global node (where $m = 0, 1, \ldots, 2N$) is given by

$$x_m = m\frac{h}{2}.$$

Note that nodes with even indices $m = 2n$ ($n = 0..N$) lie at the boundaries between elements (or domain ends), while nodes with odd indices $m = 2n+1$ ($n = 0..N-1$) lie at the midpoints of the elements.

An element e (where $e = 1, 2, \ldots, N$) is defined by three consecutive global nodes: a start node $i = 2(e - 1)$, a midpoint node $j = 2e - 1$, and an end node $k = 2e$. The element thus spans the physical interval $[x_i, x_k]$.

We seek to determine the approximate vertical position y at each global node m. Let $y_m \approx y(x_m)$ denote the nodal value at node m. These y_m values are the fundamental variables in our discretized problem. The boundary conditions fix the values at the first and last nodes:

$$y_0 = y(x_0) = y(0) = 0 \quad , \quad y_{2N} = y(x_{2N}) = y(x_f) = y_f.$$

The actual unknowns to be solved for are the values at the interior nodes ($m = 1, 2, \ldots, 2N-1$). We collect these unknown nodal values into the vector of degrees of freedom:

$$\mathbf{y}_{\text{int}} = [y_1, y_2, \ldots, y_{2N-1}]^T \in \mathbb{R}^{2N-1}.$$

1.1 Local Coordinate System

To define approximations consistently within each element, it is convenient to map the physical coordinates x belonging to an element e (i.e., $x \in [x_i, x_k]$) to a dimensionless local coordinate $\xi \in [-1, 1]$. The mapping places the local origin $\xi = 0$ at the element's midpoint node x_j:

$$x(\xi) := \frac{x_i + x_k}{2} + \frac{x_k - x_i}{2}\xi = x_j + \frac{h}{2}\xi.$$

Here, the local coordinate $\xi = -1$ corresponds to the element's start node x_i, $\xi = 0$ corresponds to the midpoint node x_j, and $\xi = 1$ corresponds to the end node x_k.

The Jacobian of this transformation relates the physical and local differentials:

$$J = \frac{dx}{d\xi} = \frac{x_k - x_i}{2} = \frac{h}{2}.$$

Hence, the differential transformation is

$$dx = J d\xi = \frac{h}{2} d\xi.$$

1.2 Quadratic Shape Functions for Approximating $y(x)$ and $y'(x)$

Within element e, let the nodal values corresponding to its start, mid, and end nodes be collected in the element nodal vector $\mathbf{y}_e = [y_i, y_j, y_k]^T$, where i, j, k are the global indices for element e. We approximate the function $y(x)$ within this element using an interpolation $y_h(x)$ based on these nodal values and quadratic shape functions of the local coordinate ξ:

$$y_h\left(x(\xi)\right) = N_1(\xi)y_i + N_2(\xi)y_j + N_3(\xi)y_k.$$

The quadratic shape functions $N_1(\xi), N_2(\xi), N_3(\xi)$ must satisfy the interpolation property

$$N_m(\xi_n) = \delta_{mn} \quad \text{for } m, n = 1, 2, 3,$$

where we associate local node numbers $1, 2, 3$ with local coordinates $\xi_1 = -1, \quad \xi_2 = 0, \quad \xi_3 = 1$, and where δ_{mn} denotes the Kronecker delta. This property ensures that the approximation y_h exactly matches the nodal values at the element's nodes: $y_h(x_i) = y_i$, $y_h(x_j) = y_j$, $y_h(x_k) = y_k$.

The unique quadratic polynomials satisfying these conditions are the Lagrange polynomials on $[-1, 1]$ for nodes at $-1, 0, 1$:

$$\begin{aligned} N_1(\xi) &= \frac{(\xi - 0)(\xi - 1)}{(-1-0)(-1-1)} = \frac{1}{2}\xi(\xi - 1), \\ N_2(\xi) &= \frac{(\xi - (-1))(\xi - 1)}{(0 - (-1))(0 - 1)} = \frac{(\xi + 1)(\xi - 1)}{-1} = 1 - \xi^2, \\ N_3(\xi) &= \frac{(\xi - (-1))(\xi - 0)}{(1 - (-1))(1 - 0)} = \frac{1}{2}\xi(\xi + 1). \end{aligned}$$

The element approximation can be written compactly using vector notation:

$$y_h\left(x(\xi)\right) = \mathbf{N}(\xi) \cdot \mathbf{y}_e,$$

where $\mathbf{N}(\xi) = [N_1(\xi),\ N_2(\xi),\ N_3(\xi)]$ is the vector of shape functions.

Since the time functional $T[y(x)]$ also depends on the derivative $y'(x)$, we approximate this with the derivative of the interpolation, $y_h'(x)$. Using the chain rule:

$$y_h'(x) = \frac{dy_h}{dx} = \frac{dy_h}{d\xi}\frac{d\xi}{dx}.$$

The derivative with respect to the local coordinate is

$$\frac{dy_h}{d\xi} = \frac{d}{d\xi}(\mathbf{N}(\xi) \cdot \mathbf{y}_e) = \left(\frac{d\mathbf{N}}{d\xi}\right) \cdot \mathbf{y}_e,$$

where $\frac{d\mathbf{N}}{d\xi} = \left[\xi - \frac{1}{2},\ -2\xi,\ \xi + \frac{1}{2}\right]$. Using the inverse Jacobian $\frac{d\xi}{dx} = 1/J = 2/h$, the approximation for the physical derivative becomes:

$$y_h'(x(\xi)) = \left(\frac{d\mathbf{N}}{d\xi} \cdot \mathbf{y}_e\right)\frac{2}{h} = \frac{2}{h}\left(\left[\xi - \frac{1}{2},\ -2\xi,\ \xi + \frac{1}{2}\right]\mathbf{y}_e\right).$$

1.3 Discretized Functional

For numerical optimization, we minimize the functional T^* which omits the constant factor $1/\sqrt{2g}$ from the original total time functinoal:

$$T^*[y(x)] = \int_0^{x_f} \sqrt{\frac{1 + (y'(x))^2}{y(x)}}dx.$$

We approximate this functional by replacing the exact function $y(x)$ and its derivative $y'(x)$ with their finite element approximations $y_h(x)$ and $y_h'(x)$. The integral over the full domain $[0, x_f]$ is computed as the sum of integrals over each element e:

$$T^*[y(x)] \approx T_h(\mathbf{y}) = \sum_{e=1}^{N} T_e^*(\mathbf{y}_e)$$

where T_e^* is the contribution from element e, whose associated global nodes are i, j, k:

$$T_e^*(\mathbf{y}_e) = \int_{x_i}^{x_k} \sqrt{\frac{1 + (y_h'(x))^2}{y_h(x)}} dx.$$

The total approximate functional T_h is now expressed solely in terms of the nodal values $\mathbf{y}_m$, specifically the unknown ones contained in the vector $\mathbf{y}$. The next step involves evaluating these element integrals T_e^* using numerical quadrature.