
Práctica 3: El problema de la braquistócrona usando técnicas de optimización

Álvaro Ortiz Villa y Erik Altelarrea Ferré

Programación Matemática (G. Matemáticas, FME). Diciembre 2021

Índice

1. Formulación matemática del problema	3
1.1. Primera variante del problema	3
1.2. Segunda variante del problema	4
1.2.1. Demostración de la convexidad del problema	4
1.3. Tercera variante del problema	6
2. Modelo AMPL desarrollado	6
2.1. Primera variante del problema	6
2.1.1. Partición uniforme	6
2.1.2. Partición no uniforme	7
2.2. Segunda variante del problema	7
2.2.1. Partición uniforme	7
2.2.2. Partición no uniforme	8
2.3. Tercera variante del problema	9
2.4. Ejecutable	9
3. Resultados obtenidos	10
3.1. Primera variante del problema	10
3.2. Segunda variante del problema	11
3.3. Tercera variante del problema	11
4. Comparación y conclusiones	12
4.1. Variantes con partición uniforme	12
4.2. Partición uniforme vs partición no uniforme	13
4.3. Algunas consideraciones	15

1. Formulación matemática del problema

Dados dos puntos diferentes de \mathbb{R}^2 , $O = (0, 0)$ y $F = (a, b)$, buscamos una discretización de la curva braquistócrona $f: \mathbb{R} \rightarrow \mathbb{R}$ que recorrería un objeto para ir de O a F suponiendo que únicamente actúa la fuerza de la gravedad y que no hay fricción. En las tres versiones consideraremos $(a, b) = (1, 1)$, es por ello que de la formulación del problema del enunciado podemos escribir la función objetivo como sigue:

$$z = \min_{f(x)} \frac{1}{\sqrt{2g}} \int_0^a \sqrt{\frac{1 + f'(x)^2}{f(x)}} dx = \min_{f(x)} \frac{1}{\sqrt{2g}} \int_0^1 \sqrt{\frac{1 + f'(x)^2}{f(x)}} dx$$

si expresamos la integral mediante una suma de Riemann asociada a la partición $\mathcal{P}_n = \{x_0, \dots, x_n\}$ del intervalo $[0, 1]$, donde $x_0 = 0 < x_1 = \frac{1}{n} < \dots < x_{n-1} = \frac{n-1}{n} < x_n = 1$, obtenemos:

$$\int_0^1 \sqrt{\frac{1 + f'(x)^2}{f(x)}} dx = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^n \sqrt{\frac{1 + f'(\frac{i}{n})^2}{f(\frac{i}{n})}}$$

usando la aproximación

$$f'(\frac{i}{n}) = f'(x_i) = \lim_{h \rightarrow 0} \frac{f(\frac{i}{n} + h) - f(\frac{i}{n})}{h} \approx \frac{f(\frac{i}{n} + \frac{1}{n}) - f(\frac{i}{n})}{\frac{1}{n}} = \frac{f(x_{i+1}) - f(x_i)}{\frac{1}{n}} \quad \forall n \text{ suficientemente grande,}$$

tenemos:

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^n \sqrt{\frac{1 + f'(\frac{i}{n})^2}{f(\frac{i}{n})}} &\approx \frac{1}{n} \sum_{i=1}^n \sqrt{\frac{1 + \left(\frac{f(x_i) - f(x_{i-1}))}{\frac{1}{n}}\right)^2}{f(x_{i-1})}} = \frac{1}{n} \sum_{i=1}^n \sqrt{\frac{(\frac{1}{n})^2 + (f(x_i) - f(x_{i-1}))^2}{f(x_{i-1}) (\frac{1}{n})^2}} = \\ &= \sum_{i=1}^n \sqrt{\frac{(\frac{1}{n})^2 + (f(x_i) - f(x_{i-1}))^2}{f(x_{i-1})}} = \sum_{i=1}^n \sqrt{\frac{(x_i - x_{i-1})^2 + (f(x_i) - f(x_{i-1}))^2}{f(x_{i-1})}} = \\ &= \sum_{i=1}^n \sqrt{\frac{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}{y_{i-1}}} \end{aligned}$$

Puesto que para resolver el problema buscamos las x_i y/o y_i ($i \in \{0, \dots, n\}$) que minimizan la función objetivo y $\frac{1}{\sqrt{2g}} > 0$, podemos omitir esta constante en la formulación del problema ya que las x y/o y que minimizan $\frac{1}{\sqrt{2g}} \int_0^a \sqrt{\frac{1 + f'(x)^2}{f(x)}} dx$ son las mismas que minimizan $\int_0^a \sqrt{\frac{1 + f'(x)^2}{f(x)}} dx$.

1.1. Primera variante del problema

En esta variante del problema consideraremos las abscisas x_i como parámetros (valores fijados) y las ordenadas y_i como variables del problema de optimización, hallaremos $y = f(x)$. Para evitar dividir entre 0, junto a las restricciones de la trayectoria impondremos $y_i \geq \text{tol} \quad \forall i \in \{0, \dots, n\}$. Ello también responde a una cuestión física, dado que la fuerza de la gravedad es la única que actúa sobre el objeto y por tanto las ordenadas y_i deben ser positivas.

$$(P) \left\{ \begin{array}{l} \min_y \sum_{i=1}^n \sqrt{\frac{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}{y_{i-1}}} \\ \text{s.a.} \\ y_0 = \text{tol} \\ y_n = b \\ y_i \geq \text{tol} \quad \forall i \in \{0, \dots, n\} \end{array} \right.$$

1.2. Segunda variante del problema

En esta variante del problema consideraremos las ordenadas y_i como parámetros y las abscisas x_i como variables, de tal modo que hallaremos la función inversa, $x = f^{-1}(y)$.

Para garantizar la existencia de dicha inversa, ésta debe ser biyectiva así que impondremos la inyectividad de la función mediante la restricción $x_i - x_{i-1} \geq \text{tol} \quad \forall i \in \{1, \dots, n\}$. La exhaustividad se soluciona redefiniendo el conjunto de llegada de la función inversa. Finalmente, para evitar dividir entre 0 y por la cuestión física mencionada anteriormente, definiremos los parámetros $y_i \geq \text{tol} \quad \forall i \in \{0, \dots, n\}$. Por la misma razón física, impondremos que las variables x_i sean positivas.

$$(P) \left\{ \begin{array}{l} \min_x \sum_{i=1}^n \sqrt{\frac{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}{y_{i-1}}} \\ \text{s.a.} \\ x_0 = 0 \\ x_n = a \\ x_i \geq 0 \quad \forall i \in \{0, \dots, n\} \\ x_i - x_{i-1} \geq \text{tol} \quad \forall i \in \{1, \dots, n\} \end{array} \right.$$

1.2.1. Demostración de la convexidad del problema

Veamos, a continuación, que esta variante proporciona un problema de optimización convexo, i.e. la función objetivo es convexa y la región factible es convexa.

Al ser las restricciones de este problema funciones lineales, la región factible es convexa.

Recordamos que una función, f , es convexa $\iff f(tx + (1-t)y) \leq tf(x) + (1-t)f(y) \quad \forall t \in [0, 1], \forall x, y \in \text{Dom}(f)$.

Para verlo emplearemos la caracterización de teoría para funciones convexas: $f \in \mathcal{C}^2$ es convexa $\iff \nabla^2 f \succeq^1 0$.

Con el fin de simplificar la demostración, definimos

$$h_i(x_{i-1}, x_i) = \sqrt{\frac{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}{y_{i-1}}}$$

de tal modo que la función objetivo es $\sum_{i=1}^n h_i$, que naturalmente es \mathcal{C}^2 en ser suma y composición de funciones \mathcal{C}^2 . Puesto que en teoría demostramos que la suma de funciones convexas es convexa, nos podemos reducir, sin pérdida de generalidad, a ver que h_i es convexa $\forall i \in \{1, \dots, n\}$. Para ello

¹Empleamos este operador para indicar que una matriz es semidefinida positiva; es decir, $M \succeq 0 \iff \forall x \in \mathbb{R}^n \quad x^\top M x \geq 0$.

veremos que $\nabla^2 h_i \succeq 0 \forall i \in \{1, \dots, n\}$. Claramente tenemos que

$$\nabla h_i(x_{i-1}, x_i) = \left(\frac{x_{i-1} - x_i}{\sqrt{y_{i-1}} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}, \frac{x_i - x_{i-1}}{\sqrt{y_{i-1}} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}} \right)^\top$$

luego

$$\nabla^2 h_i(x_{i-1}, x_i) = \begin{pmatrix} \frac{(y_i - y_{i-1})^2}{\sqrt{y_{i-1}} ((x_i - x_{i-1})^2 + (y_i - y_{i-1})^2) \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}} & -\frac{(y_i - y_{i-1})^2}{\sqrt{y_{i-1}} ((x_i - x_{i-1})^2 + (y_i - y_{i-1})^2) \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}} \\ -\frac{(y_i - y_{i-1})^2}{\sqrt{y_{i-1}} ((x_i - x_{i-1})^2 + (y_i - y_{i-1})^2) \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}} & \frac{(y_i - y_{i-1})^2}{\sqrt{y_{i-1}} ((x_i - x_{i-1})^2 + (y_i - y_{i-1})^2) \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}} \end{pmatrix}$$

Veamos que es semidefinida positiva, para ello emplearemos la caracterización vista en teoría: una matriz es semidefinida positiva \iff todos sus menores² son ≥ 0 :

$$\Delta_1 = \frac{(y_i - y_{i-1})^2}{\sqrt{y_{i-1}} ((x_i - x_{i-1})^2 + (y_i - y_{i-1})^2) \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}} \geq 0 \forall (x_{i-1}, x_i) \in \mathbb{R}^2.$$

$$\Delta_2 = \left(\frac{(y_i - y_{i-1})^2}{\sqrt{y_{i-1}} ((x_i - x_{i-1})^2 + (y_i - y_{i-1})^2) \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}} \right)^2 - \left(\frac{(y_i - y_{i-1})^2}{\sqrt{y_{i-1}} ((x_i - x_{i-1})^2 + (y_i - y_{i-1})^2) \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}} \right)^2 = 0 \geq 0 \forall (x_{i-1}, x_i) \in \mathbb{R}^2.$$

$$\Delta_3 = \frac{(y_i - y_{i-1})^2}{\sqrt{y_{i-1}} ((x_i - x_{i-1})^2 + (y_i - y_{i-1})^2) \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}} \geq 0 \forall (x_{i-1}, x_i) \in \mathbb{R}^2.$$

$$\implies \nabla^2 h_i \succeq 0.$$

Siguiendo la notación, Δ_1 hace referencia al primer menor, a la entrada $(1, 1)$ de $\nabla^2 h_i$, Δ_3 a la entrada $(2, 2)$ de la misma, mientras que Δ_2 referencia el determinante de dicha matriz.

Alternativamente, sea $v = (v_1, v_2)^\top \in \mathbb{R}^2$,

$$v^\top \nabla^2 h_i v = \frac{(y_i - y_{i-1})^2}{\sqrt{y_{i-1}} ((x_i - x_{i-1})^2 + (y_i - y_{i-1})^2) \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}} (v_1 - v_2)^2 \geq 0 \forall v \in \mathbb{R}^2$$

$$\implies \nabla^2 h_i \succeq 0.$$

Así, queda demostrado que es un problema de optimización convexo. ■

²Nótese que no hablamos únicamente de los menores principales.

1.3. Tercera variante del problema

En esta variante consideraremos tanto x_i como y_i como variables. Impondremos también condiciones de inyectividad de f y de f^{-1} , esta vez, con una tolerancia mayor a la anterior, $1e - 7$. Además, para evitar dividir entre 0 y por las cuestiones físicas mencionadas, impondremos $y_i \geq \text{tol}$ y $x_i \geq 0 \forall i \in \{0, \dots, n\}$.

$$(P) \left\{ \begin{array}{l} \min_{x,y} \sum_{i=1}^n \sqrt{\frac{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}{y_{i-1}}} \\ \text{s.a.} \\ x_0 = 0 \\ x_n = a \\ y_0 = \text{tol} \\ y_n = b \\ x_i \geq 0 \forall i \in \{0, \dots, n\} \\ y_i \geq \text{tol} \forall i \in \{0, \dots, n\} \\ x_i - x_{i-1} \geq 1e - 7 \forall i \in \{1, \dots, n\} \\ y_i - y_{i-1} \geq 1e - 7 \forall i \in \{1, \dots, n\} \end{array} \right.$$

Antes de presentar los modelos AMPL elaborados y los resultados de ejemplo obtenidos, podemos imaginar que la última variante proporcionará posiblemente una mejor aproximación a la cicloide invertida ya que considera tanto x_i como y_i como variables sin imponer cómo debe ser la discretización del segmento.

2. Modelo AMPL desarrollado

En esta sección proporcionaremos los códigos desarrollados en cada una de las versiones así como el ejecutable. En las dos primeras variantes del problema hemos propuesto dos discretizaciones diferentes los parámetros. La primera corresponde a una partición uniforme, mientras que la segunda a una partición no uniforme que concentra un mayor número de puntos de la partición en el inicio del intervalo.

2.1. Primera variante del problema

2.1.1. Partición uniforme

Código 1 – ModeloA.MOD

```

1 #Modelo A Braquistócrona
2
3 param n, integer;
4 param a;
5 param b;
6 param tol > 0;
7
8 #Discretización de los parámetros:
9 param x{i in 0 .. n} = a*i/n;
10
```

```

11 #Declaración de las variables:
12 var y{i in 0 .. n} >= tol;
13
14 # Función objetivo
15 minimize total:
16 sum {i in 1 .. n}
17 sqrt(((x[i] - x[i - 1])**2 + (y[i] - y[i - 1])**2)/y[i - 1]);
18
19 # Restricciones de la trayectoria
20 subject to primera: y[0] = tol;
21 subject to segunda: y[n] = b;

```

2.1.2. Partición no uniforme

Para la primera variante del problema con partición no uniforme hemos tenido que imponer una tercera restricción adicional en relación a la inyectividad de la función f .

Código 2 – ModeloA NU.MOD

```

1 #Modelo A Braquistócrona
2
3 param n, integer;
4 param a;
5 param b;
6 param tol > 0;
7
8 #Discretización de los parámetros:
9 param x{i in 0 .. n} = a*(i/n)**2;
10
11 #Declaración de las variables:
12 var y{i in 0 .. n} >= tol;
13
14 # Función objetivo
15 minimize total:
16 sum {i in 1 .. n}
17 sqrt(((x[i] - x[i - 1])**2 + (y[i] - y[i - 1])**2)/y[i - 1]);
18
19 # Restricciones de la trayectoria
20 subject to primera: y[0] = tol;
21 subject to segunda: y[n] = b;
22
23 # Restricciones de inyectividad
24 subject to tercera {i in 1 .. n}: y[i] - y[i - 1] >= 1e-7;

```

2.2. Segunda variante del problema

2.2.1. Partición uniforme

Código 3 – ModeloB.MOD

```

1 #Modelo B Braquistócrona
2
3 param n, integer;
4 param a;
5 param b;
6 param tol > 0;
7
8 #Discretización de los parámetros:
9 param y{i in 0 .. n} = b*i/n + tol;
10
11 #Declaración de las variables:
12 var x{i in 0 .. n} >= 0;
13
14 # Función objetivo
15 minimize total:
16 sum {i in 1 .. n}
17 sqrt(((x[i] - x[i - 1])**2 + (y[i] - y[i - 1])**2)/y[i - 1]);
18
19 # Restricciones de la trayectoria
20 subject to primera: x[0] = 0;
21 subject to segunda: x[n] = a;
22
23 # Restricciones de inyectividad
24 subject to tercera {i in 1 .. n}: x[i] - x[i - 1] >= tol;

```

2.2.2. Partición no uniforme

Código 4 – ModeloB NU.MOD

```

1 #Modelo B Braquistócrona
2
3 param n, integer;
4 param a;
5 param b;
6 param tol > 0;
7
8 #Discretización de los parámetros:
9 param y{i in 0 .. n} = b*(i/n)**2 + tol;
10
11 #Declaración de las variables:
12 var x{i in 0 .. n} >= 0;
13
14 # Función objetivo
15 minimize total:
16 sum {i in 1 .. n}
17 sqrt(((x[i] - x[i - 1])**2 + (y[i] - y[i - 1])**2)/y[i - 1]);
18
19 # Restricciones de la trayectoria
20 subject to primera: x[0] = 0;

```



```

21 subject to segunda: x[n] = a;
22
23 # Restricciones de inyectividad
24 subject to tercera {i in 1 .. n}: x[i] - x[i - 1] >= tol;

```

2.3. Tercera variante del problema

Código 5 – ModeloC.MOD

```

1 #Modelo C Braquistócrona
2
3 param n, integer;
4 param a;
5 param b;
6 param tol > 0;
7
8 #Declaración de las variables:
9 var x{i in 0 .. n} >= 0;
10 var y{i in 0 .. n} >= tol;
11
12 # Función objetivo
13 minimize total:
14 sum {i in 1 .. n}
15 sqrt(((x[i] - x[i - 1])**2 + (y[i] - y[i - 1])**2)/y[i - 1]);
16
17 # Restricciones de la trayectoria
18 subject to primera: x[0] = 0;
19 subject to segunda: x[n] = a;
20 subject to tercera: y[0] = tol;
21 subject to cuarta: y[n] = b;
22
23 # Restricciones de inyectividad
24 subject to quinta {i in 1 .. n}: x[i] - x[i - 1] >= 1e-7;
25 subject to sexta {i in 1 .. n}: y[i] - y[i - 1] >= 1e-7;

```

2.4. Ejecutable

Código 6 – braquistocrona.RUN

```

1 # Borrar los anterior comandos en AMPL
2 reset;
3
4 option log_file output.dat;
5
6 # Indicación a AMPL de que se quiere utilizar como solver
7 option solver minos;
8 option minos_options 'superbasics_limit= 1000';
9

```

```

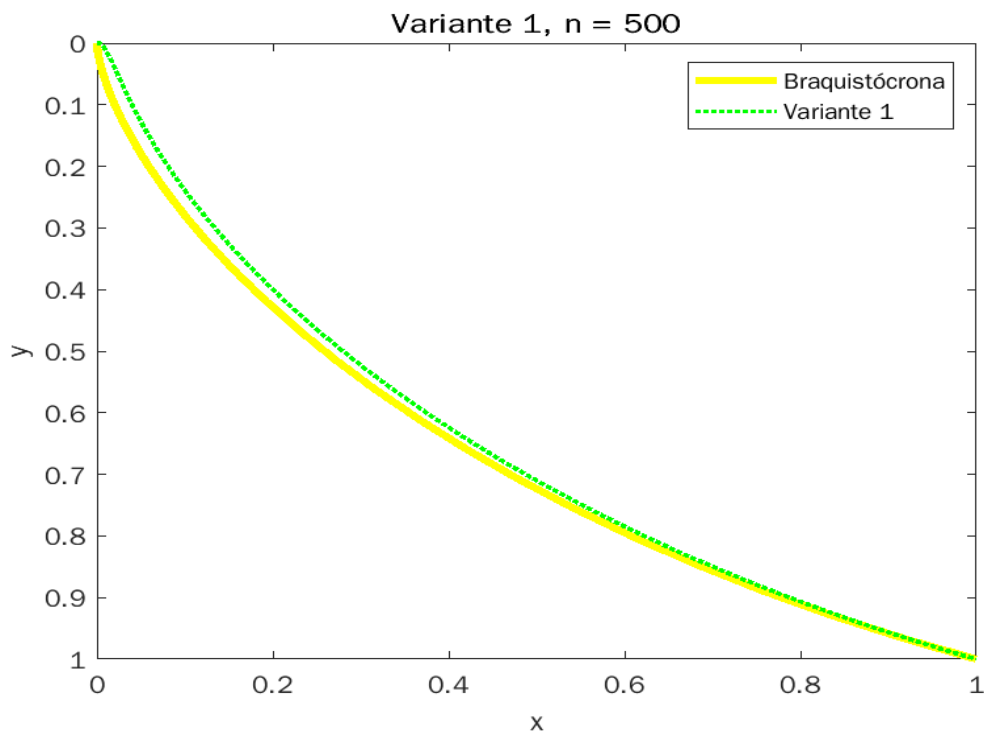
10 # Cargar el modelo
11 model ModeloA.mod;
12 # Cargar los datos
13 data datos.dat;
14
15 #Resolver
16 solve;
17 display x, y;
18 printf "\n";
19
20 option log_file '';
21 close output.dat;

```

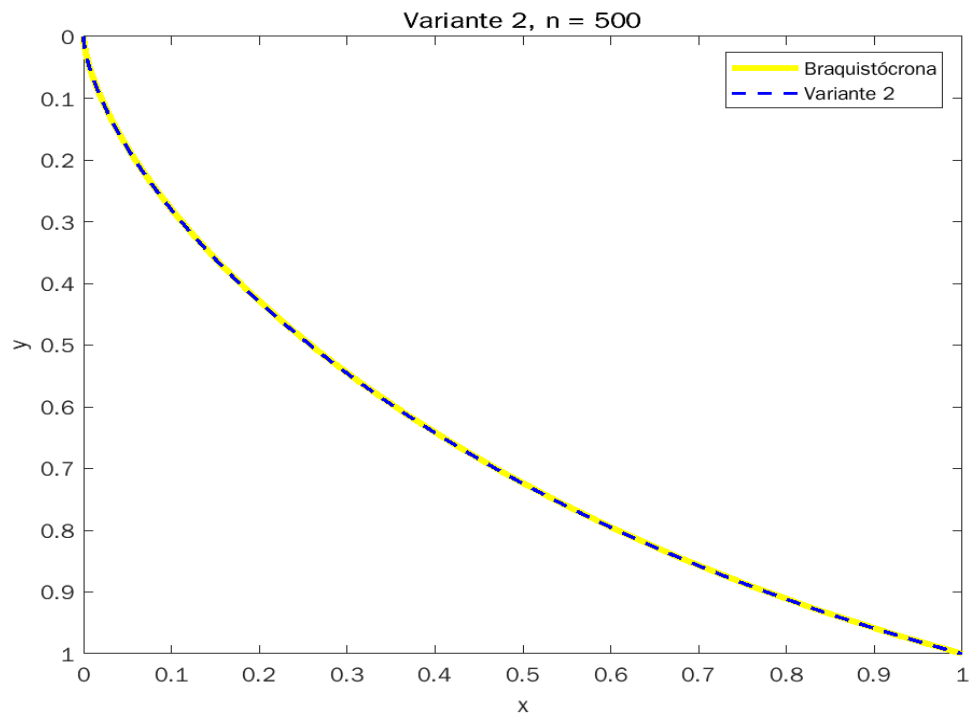
3. Resultados obtenidos

En esta sección proporcionaremos los resultados de ejemplo obtenidos, en forma de gráfico representado por Matlab, de cada una de las variantes junto con la representación de la braquistócrona o cicloide invertida, la curva que pretendíamos aproximar en esta práctica. En primer lugar, se representa cada variante del problema con una partición uniforme de 500 puntos. En segundo lugar, se presentan unos gráficos a modo de comparación en los que aparecen las curvas obtenidas con cada una de las 3 variantes y la braquistócrona para distintos valores de n . Finalmente estudiamos la mejoría al tomar una partición no uniforme.

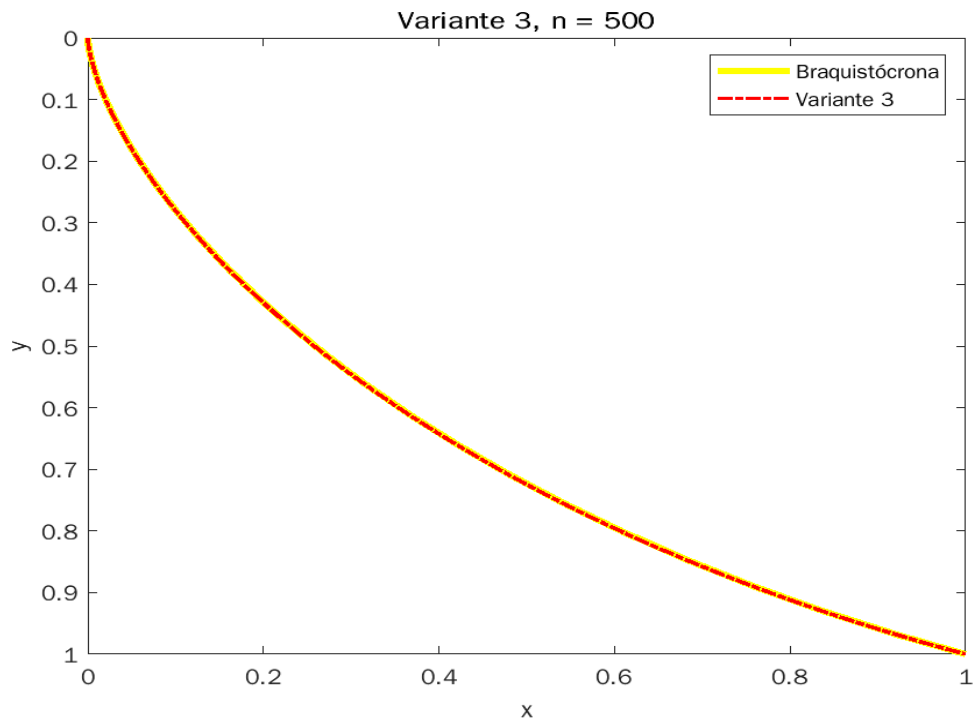
3.1. Primera variante del problema



3.2. Segunda variante del problema



3.3. Tercera variante del problema



4. Comparación y conclusiones

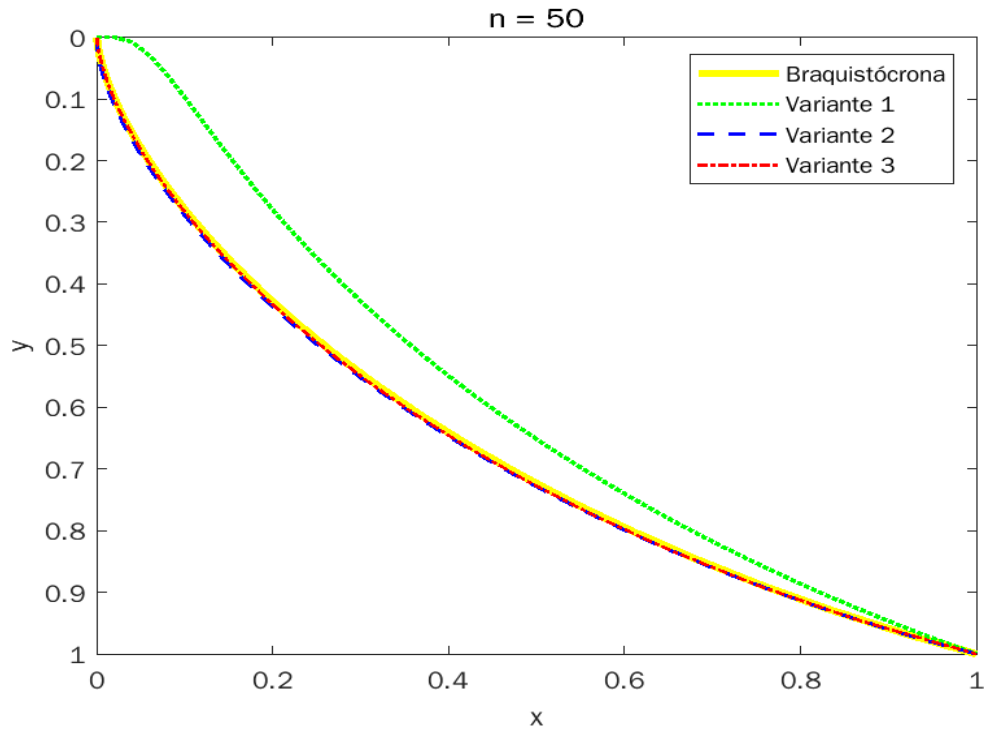
4.1. Variantes con partición uniforme

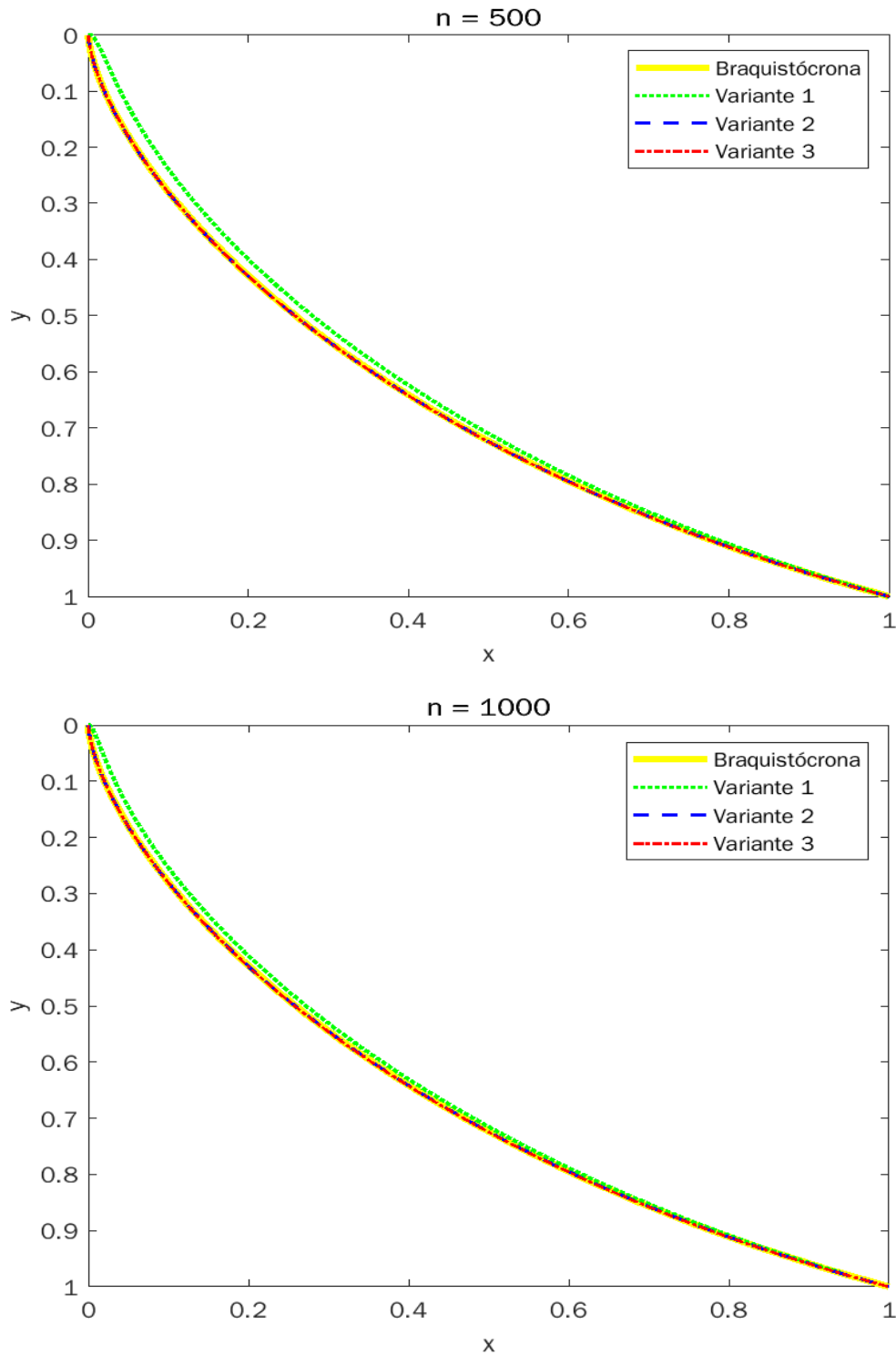
Los siguientes gráficos muestran, para diferentes valores de n , los resultados obtenidos en función de la variante usada. Se muestra también el valor exacto de la braquistócrona para tener una referencia en cuanto a la calidad de la aproximación.

De entrada, se cumple que las curvas se ajustan mejor a una cicloide invertida cuanto más fina es la partición del segmento. Para las distintas particiones se observa una notable mejoría de la aproximación de la segunda variante respecto de la primera. Sabemos de teoría que la convexidad del problema de optimización conlleva que los óptimos locales sean óptimos globales. Ello facilita la convergencia hacia el punto óptimo porque los métodos de optimización que hemos empleado describen el óptimo mediante la condición necesaria de primer orden. Ahora bien, si el problema no es convexo entonces dicha condición sólo es necesaria pero no es suficiente.

Cabe añadir que la segunda y tercera variante aproximan mejor que la primera, y es para particiones poco finas donde apreciamos que la tercera variante se ajusta ligeramente mejor a la segunda. Esta observación cumple nuestra hipótesis inicial y parece responder al hecho de que en el tercer modelo la discretización de los puntos es endógena al problema de optimización. Es decir, a diferencia de las dos formulaciones anteriores, en la última no se ha impuesto nada sobre los valores de la partición al considerar las x_i e y_i como variables del problema y no como parámetros. Sin embargo, en esta variante tenemos el doble de variables y por lo tanto el problema a resolver es significativamente mayor. A modo de ejemplo, para $n = 1000$ la segunda variante encuentra el óptimo en 1486 iteraciones mientras que la tercera variante en 6330.

Por otro lado, las dos primeras variantes, y en especial la primera, no parecen ajustar bien la curva en el comienzo del segmento, aspecto que nos motiva a considerar discretizaciones no uniformes en el siguiente apartado.

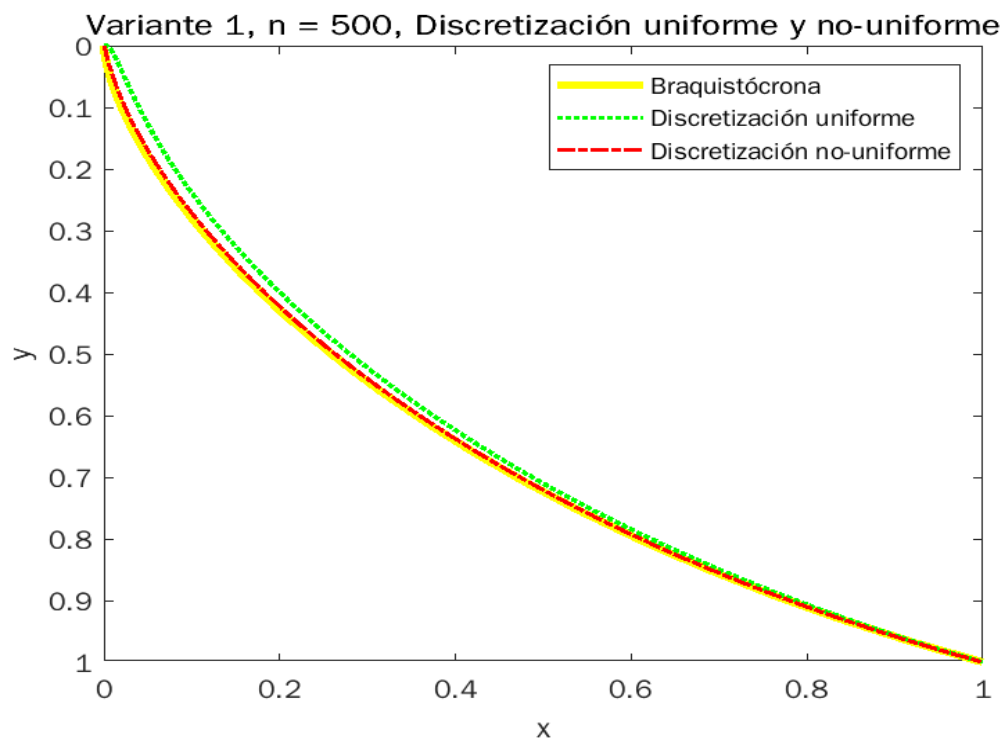
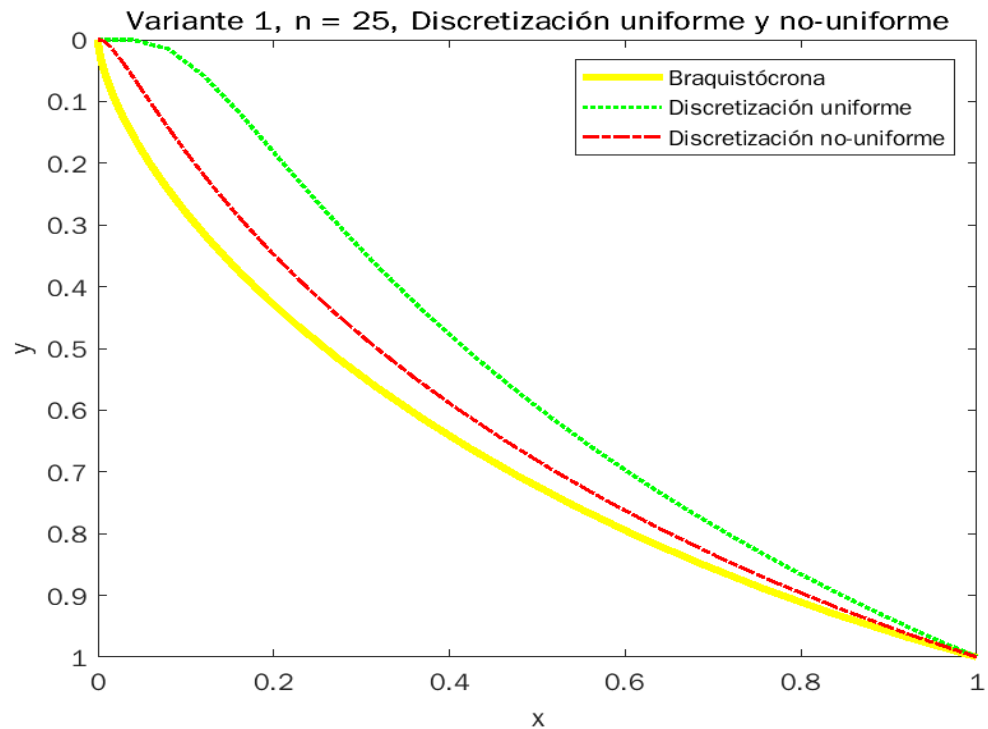


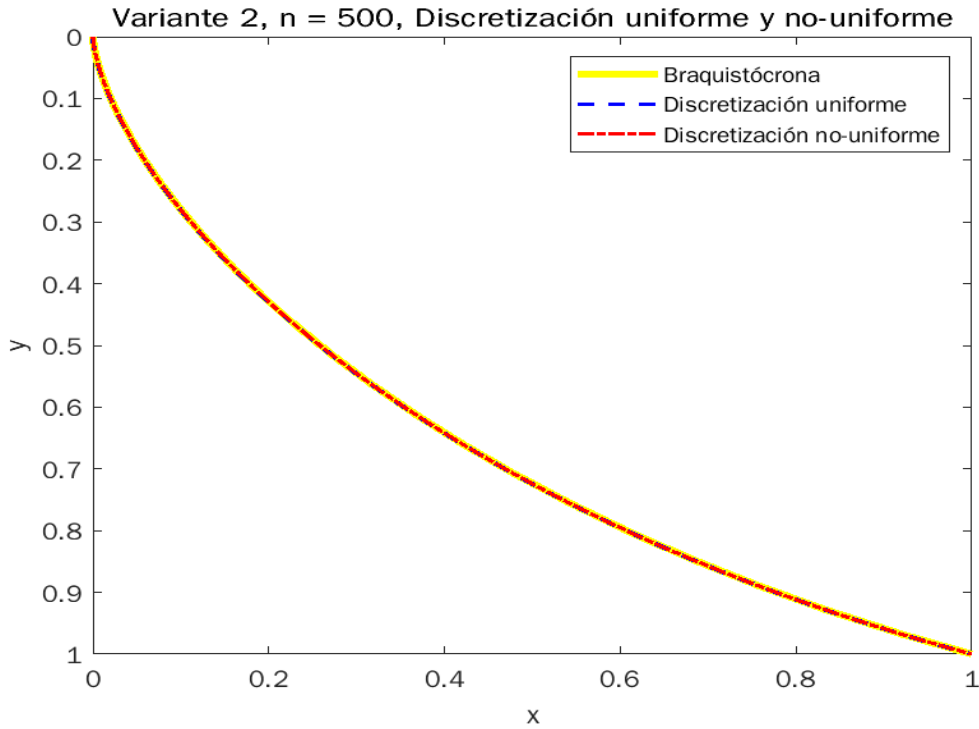
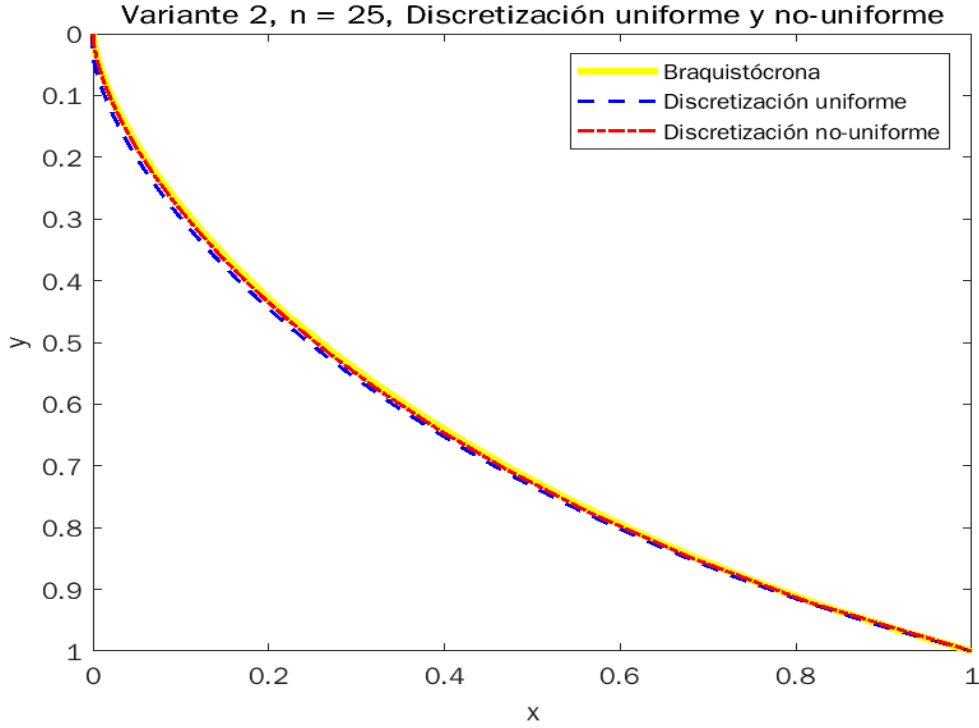


4.2. Partición uniforme vs partición no uniforme

En esta sección mostramos el resultado obtenido en las variantes 1 y 2 del problema en función de la partición. Para poder apreciar bien la diferencia hemos usado valores de n pequeños. En efecto, observamos que una mayor concentración de puntos en el inicio del segmento produce en ambas variantes una mejor aproximación a una cicloide invertida. Esto es patente para particiones poco finas en ambas variantes e incluso para particiones finas de la primera variante, mientras que para

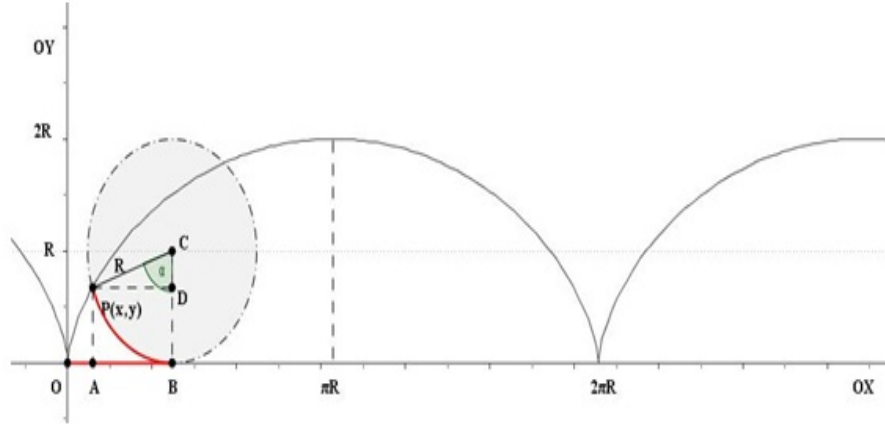
particiones finas de la segunda formulación obtenemos una curva similar a la que obtenemos con una discretización uniforme.





4.3. Algunas consideraciones

- En la segunda y tercera variante del problema se ha impuesto la condición de inyectividad: $x_i - x_{i-1} \geq \text{tol} \forall i \in \{1, \dots, n\}$ y $y_i - y_{i-1} \geq 1e - 7 \forall i \in \{1, \dots, n\}$ respectivamente, de manera que la función resultante es inyectiva. Dicha función constituirá una aproximación de la cicloide en un intervalo en el que esta sea inyectiva. Como vemos en la siguiente representación de la cicloide:



mantenemos la inyectividad cuando dado un punto, A en la imagen, lo rotamos un máximo de πR radianes. Si giráramos el punto más de πR radianes, la aproximación sería incorrecta, pues tendríamos una contradicción: una función inyectiva es igual a una función no inyectiva. Para estos casos habría que emplear la primera variante, que no depende de la inyectividad de la función.

- Nuestra función objetivo, $\sum_{i=1}^n \sqrt{\frac{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}{y_{i-1}}}$, requiere $y_i > 0 \forall i \in \{0, \dots, n\}$ para estar definida, condición que se ha impuesto en las 3 variantes para evitar posibles divisiones entre 0 y argumentos de raíz cuadrada negativos.