

Universidad de Cádiz

PNLCC
Minimal Surface with Obstacle

ADRIÁN FERNÁNDEZ TEJADA

ADRIÁN SÁNCHEZ LOUREIRO

2023

INTRODUCCIÓN AL PROBLEMA	2	RESULTADOS	7
FORMULACIÓN DEL PROBLEMA	3	CAMBIANDO OBSTÁCULOS	11

PRACTICA 2



MINIMAL SURFACE WITH OBSTACLE

Introducción al problema

El objetivo de nuestro problema del Benchmark [1, 17 p. 39] es encontrar la superficie con área mínima que se encuentra sobre un obstáculo con ciertas condiciones de contorno dadas. Los autores del Benchmark han extraído el problema de [2].

En matemáticas y física, una superficie mínima es una superficie que minimiza su área mientras logra una restricción: un conjunto de puntos, o el borde de la superficie, se determina de antemano [3].

La figura 1.1 representa aproximadamente a la superficie que queremos llegar.

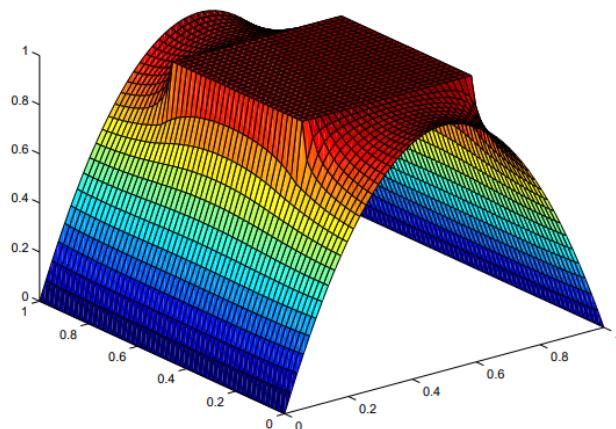


Figura 1.1

Formulación del problema

El objeto de estudio de este informe se conoce como el problema de “Plateau”, el cual pretende determinar la superficie de área mínima con una curva cerrada en \mathbb{R}^3 como frontera.

En matemáticas, el problema de Plateau [4] es mostrar la existencia de una superficie minimal con una frontera dada, un problema planteado por Lagrange en 1760. Sin embargo, fue nombrado posteriormente por Joseph Plateau quien experimentó con películas de jabón (figura 1.2).



Figura 1.2

Asumiendo que la superficie en cuestión puede ser representada por una forma no paramétrica $v : \mathbb{R}^2 \rightarrow \mathbb{R}$. Una forma no paramétrica es una técnica de modelado matemático que utiliza datos de puntos dispersos en un espacio tridimensional para construir una superficie continua sin hacer suposiciones explícitas sobre la forma funcional de la superficie [5].

Nuestra superficie es tal que $v \geq v_L$, donde v_L denota al obstáculo. Tenemos que la solución del problema [2] es aquella que minimiza la función $f : K \rightarrow \mathbb{R}$, donde:

$$f(v) = \int_{\mathcal{D}} (1 + \|\nabla v(x)\|^2)^{(1/2)} \ dx$$

sobre el conjunto convexo K , donde:

$$K = \{v \in H^1(\mathcal{D}) : v(x) = v_{\mathcal{D}}(x) \text{ para } x \in \partial\mathcal{D}, v(x) \geq v_L(x) \text{ para } x \in \mathcal{D}\}$$

donde $H^1(\mathcal{D})$ es el espacio de funciones con gradiente en $L^2(\mathcal{D})$, que a su vez es el conjunto de aplicaciones dos veces continuamente diferenciables, la función $v_{\mathcal{D}} : \partial\mathcal{D} \rightarrow \mathbb{R}$ define la información del borde y $v_L : \mathcal{D} \rightarrow \mathbb{R}$ es el obstáculo. Asumimos que $v_L \leq v_{\mathcal{D}}$ en el borde $\partial\mathcal{D}$.

Para tratar con número finito de variables en el problema, aproximaremos la superficie v definiendo una malla sobre ella. Para definir la malla seccionaremos el espacio \mathcal{D} . Primero, dados los números

$n_x, n_y \in \mathbb{N}$, dividimos el intervalo $[0, 1]$ en n_x puntos y aparte en otros n_y puntos, ambos equidistantes, para aproximar \mathcal{D} como la ‘cuadricula’ $\{0, 1, \dots, n_x\} \times \{0, 1, \dots, n_y\} \subset \mathbb{N}^2$. Hecho lo anterior, realizaremos una malla $v_{i,j}$ de $n_x \cdot n_y$ puntos equidistantes de la superficie siendo $1 \leq i \leq n_x$ y $1 \leq j \leq n_y$. Finalmente, las $n_x \cdot n_y$ variables a encontrar de nuestro problema son los valores $v_{i,j}$ para $1 \leq i \leq n_x$ y $1 \leq j \leq n_y$.

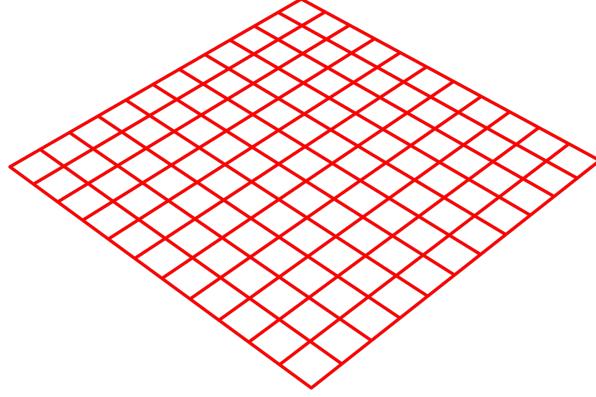


Figura 1.3: Rejilla $n_x = 10$ y $n_y = 10$

Un ejemplo de esto, se observa en la figura 1.4 y figura 1.3 con $(n_x, n_y) = (10, 10)$.

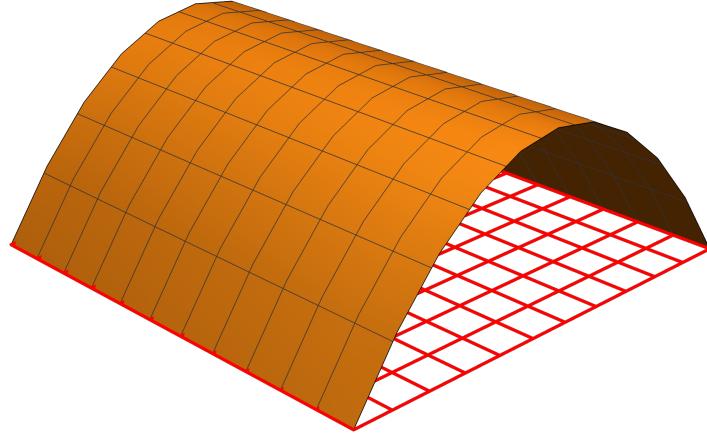


Figura 1.4: $\left\{ \left\{ \frac{i}{n_x}, \frac{j}{n_y}, v_{ij} \right\} : 0 \leq i \leq n_x, 0 \leq j \leq n_y \right\}$ para $n_x = 10$ y $n_y = 10$

Es necesario establecer condiciones analíticas tanto para la frontera v_L como para el propio obstáculo v_D que interviene en la superficie, que se definen como:

$$v_D(x, y) = \begin{cases} 1 - (2x - 1)^2, & y = 0, 1 \\ 0, & c.c. \end{cases} \quad (1.1)$$

$$v_L(x, y) = \begin{cases} 1, & \text{si } \left|x - \frac{1}{2}\right| \leq \frac{1}{4}, \quad \left|y - \frac{1}{2}\right| \leq \frac{1}{4} \\ 0, & \text{c.c.} \end{cases} \quad (1.2)$$

Para calcular numéricamente la integral $f(v)$ emplearemos la fórmula del trapecio para \mathbb{R}^2 . La fórmula extraída de [6, p. 384] para \mathbb{R} , en el intervalo $[a, b]$ con nodos $x_k = a + k \cdot H$ para $k = 0, \dots, m$ con $H = (b - a)/m$ y $m \in \mathbb{N}$ es la siguiente:

$$I_{1,m}(g) = \frac{H}{2} \left[g(x_0) + 2 \cdot g(x_1) + \dots + 2 \cdot g(x_{m-1}) + g(x_m) \right]$$

Llamemos $g(v_{i,j}) = (1 + \|\nabla v_{i,j}\|^2)^{(1/2)}$ función a integrar en \mathcal{D} . Aplicando lo anterior tenemos la discretización:

$$\begin{aligned} f(v) = \int_{\mathcal{D}} g(x) \, dx &\approx \frac{1}{4 \cdot n_x \cdot n_y} \left[g(v_{0,0}) + g(v_{0,n_y}) + g(v_{n_x,0}) + g(v_{n_x,n_y}) + 2 \cdot \sum_{j=1}^{n_y-1} g(v_{0,j}) + \right. \\ &\quad \left. 2 \cdot \sum_{i=1}^{n_x-1} g(v_{i,0}) + 2 \cdot \sum_{i=1}^{n_x-1} g(v_{i,n_y}) + 2 \cdot \sum_{j=1}^{n_y-1} g(v_{n_x,j}) + 4 \cdot \sum_{i=1}^{n_x-1} \sum_{j=1}^{n_y-1} g(v_{i,j}) \right] \end{aligned} \quad (1.3)$$

También haremos una aproximación del gradiente para los puntos $v_{i,j}$ presentes en la función $g(v_{i,j}) = (1 + \|\nabla v_{i,j}\|^2)^{(1/2)}$. Siendo la primera componente del gradiente:

$$\frac{\partial v_{i,j}}{\partial x} \approx \frac{v_{i+1,j} - v_{i-1,j}}{2 \cdot n_x}, \quad 0 \leq j \leq n_y \text{ y } 0 < i < n_x$$

Exceptuando para $i \in \{0, n_x\}$, que serían puntos con la primera coordenada que se encuentran en el borde de la superficie que queremos calcular.

$$\frac{\partial v_{0,j}}{\partial x} \approx \frac{v_{1,j} - v_{0,j}}{n_x} \quad \text{y} \quad \frac{\partial v_{n_x,j}}{\partial y} \approx \frac{v_{n_x,j} - v_{n_x-1,j}}{n_x}, \quad 0 \leq j \leq n_y$$

Para la segunda componente del gradiente:

$$\frac{\partial v_{i,j}}{\partial y} \approx \frac{v_{i,j+1} - v_{i,j-1}}{2 \cdot n_y}, \quad 0 \leq i \leq n_x \text{ y } 0 < j < n_y$$

Exceptuando para $i \in \{0, n_y\}$, que serían puntos con la segunda coordenada que se encuentran en el borde de la superficie que queremos calcular.

$$\frac{\partial v_{i,0}}{\partial y} \approx \frac{v_{i,1} - v_{i,0}}{n_y} \quad \text{y} \quad \frac{\partial v_{i,n_y}}{\partial y} \approx \frac{v_{i,n_y} - v_{i,n_y-1}}{n_y}, \quad 0 \leq i \leq n_x$$

Así pues, la función a integrar

$$g(v_{i,j}) = (1 + \|\nabla v_{i,j}\|^2)^{(1/2)} = \sqrt{1 + \frac{\partial v_{i,j}}{\partial x}^2 + \frac{\partial v_{i,j}}{\partial y}^2}$$

para $0 \leq i \leq n_x$ y $0 \leq j \leq n_y$. Tenemos en cuenta los diferentes casos del gradiente en nuestra implementación.

Finalmente, nuestras restricciones para los puntos del borde

$$\begin{aligned} v_{i,0} &= 1 - (2 * \frac{i}{n_x} - 1)^2 \quad \text{y} \quad v_{i,n_y} = 1 - (2 * \frac{i}{n_x} - 1)^2, \quad 0 \leq i \leq n_x \\ v_{0,j} &= 0 \quad \text{y} \quad v_{n_x,j} = 0, \quad 0 \leq j \leq n_y \end{aligned}$$

para los puntos $v_{i,j}$ del interior de la superficie que no están sobre el obstáculo

$$v_{i,j} \geq 0.$$

Los puntos $v_{i,j}$ sobre el obstáculo

$$v_{i,j} \geq 1.$$

Resultados

En esta sección, veremos las soluciones obtenidas y tiempos de ejecución de los distintos *solvers*.

El problema resuelto en AMPL se encuentra en la carpeta AMPL del archivo `17-SURFACE-FTA_SLA.zip`. En esta carpeta, encontramos dos subdirectorios `AMPL-2012` y `AMPL-2023`, con, respectivamente, la implementación del problema para las versiones del *software* de 2012 y 2023. En este informe, al tener la licencia limitada para la versión actual, trataremos los datos para la versión de 2012.

Además, hemos empleado en AMPL diferentes *solvers* para poder comparar los resultados. Hemos resuelto el problema para los *solver* MINOS, LOQO, IPOPT y SNOPT. El Bechmarking [1] usa las versiones de los solver: MINOS 5.5, LOQO 6.02 y SNOPT 6.1. Y nosotros hemos usado las versiones: MINOS 5.51, LOQO 6.01, IPOPT 3.0.8, SNOPT 7.2-8 y Mathematica 13.

Destacar que tanto las versiones empleadas como el ordenador por el Benchmarking [1] existentes en el momento de la publicación son anteriores a la actual. Hemos añadido estos datos en el cuadro 1.1 para reflejar una posible diferencia entre lo obtenido actualmente y en esa época.

También, para resolver los problemas en la versión 2012 de AMPL se ha utilizado un ordenador con menos especificaciones al empleado en Mathematica.

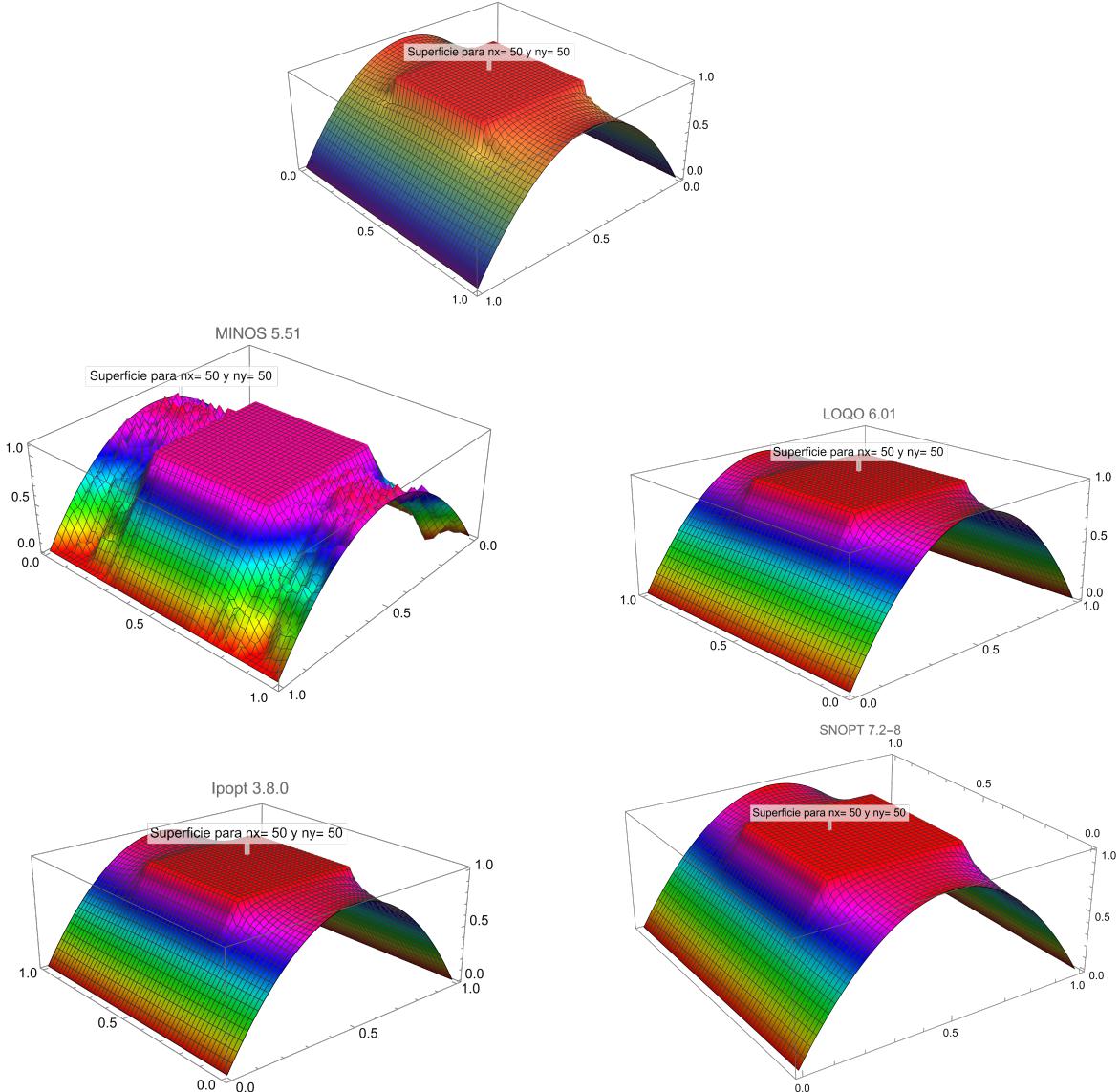
Por último, en Benchmarking [1] han triangularizado la superficie v y minimizado $f(v)$ empleando funciones a trozos lineales sobre los vértices de la triangularización. Como empleamos métodos diferentes, pero los valores n_x y n_y tienen usos similares, añadiremos estos datos a la tabla 1.1.

Solver	$n_y = 50$	$n_y = 75$	$n_y = 100$
MINOS	1.09375 s	3.796875 s	7.4375 s
Bechmarking tiempo	77.97 s	295.46 s	700.88 s
Función Objetivo	2.91639	3.05612	3.09974
Bechmarking FObj	2.51488	2.50568	2.50694
LOQO	1.671875 s	4.71875 s	8.921875 s
Bechmarking	3.99 s	6.12 s	‡
Función Objetivo	2.46163	2.46625	2.46853
Bechmarking	2.51488	2.50568	‡
SNOPT	12.328125 s	179.078125 s	496.375 s
Bechmarking	1726.62 s	‡	‡
Función Objetivo	2.46163	2.46625	2.46853
Bechmarking	2.51488	‡	‡
IPOPT	2.46875 s	6 s	10.578125 s
Función Objetivo	2.46164	2.46625	2.46854
Mathematica	8.14292 s	12.6468 s	17.9536 s
Función Objetivo	2.45378	2.45813	2.46037

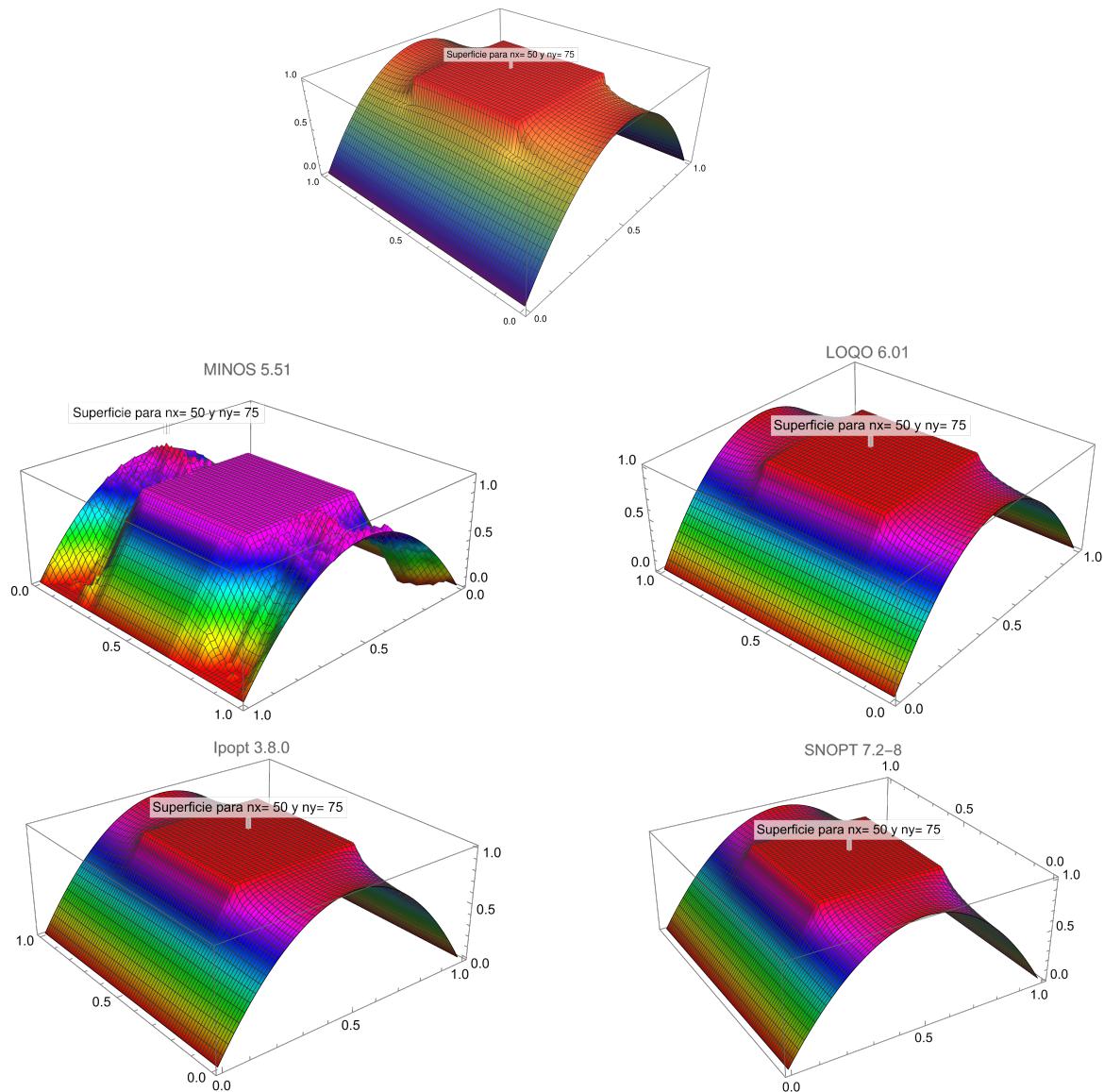
Cuadro 1.1: Resultados obtenidos fijando $n_x = 50$ para diferentes *solvers*. ‡ Tiempo máx. alcanzado

Ahora, haremos una comparación de los datos obtenidos. Podemos observar que para $n_y = 50$ en tiempo la mejor solución la da MINOS, pero obtenemos una peor gráfica y función objetivo que el resto. Así que para este n_y la mejor opción en relación función objetivo y tiempo, será LOQO. En general, todos los datos de Benchmarking son peores que los obtenidos.

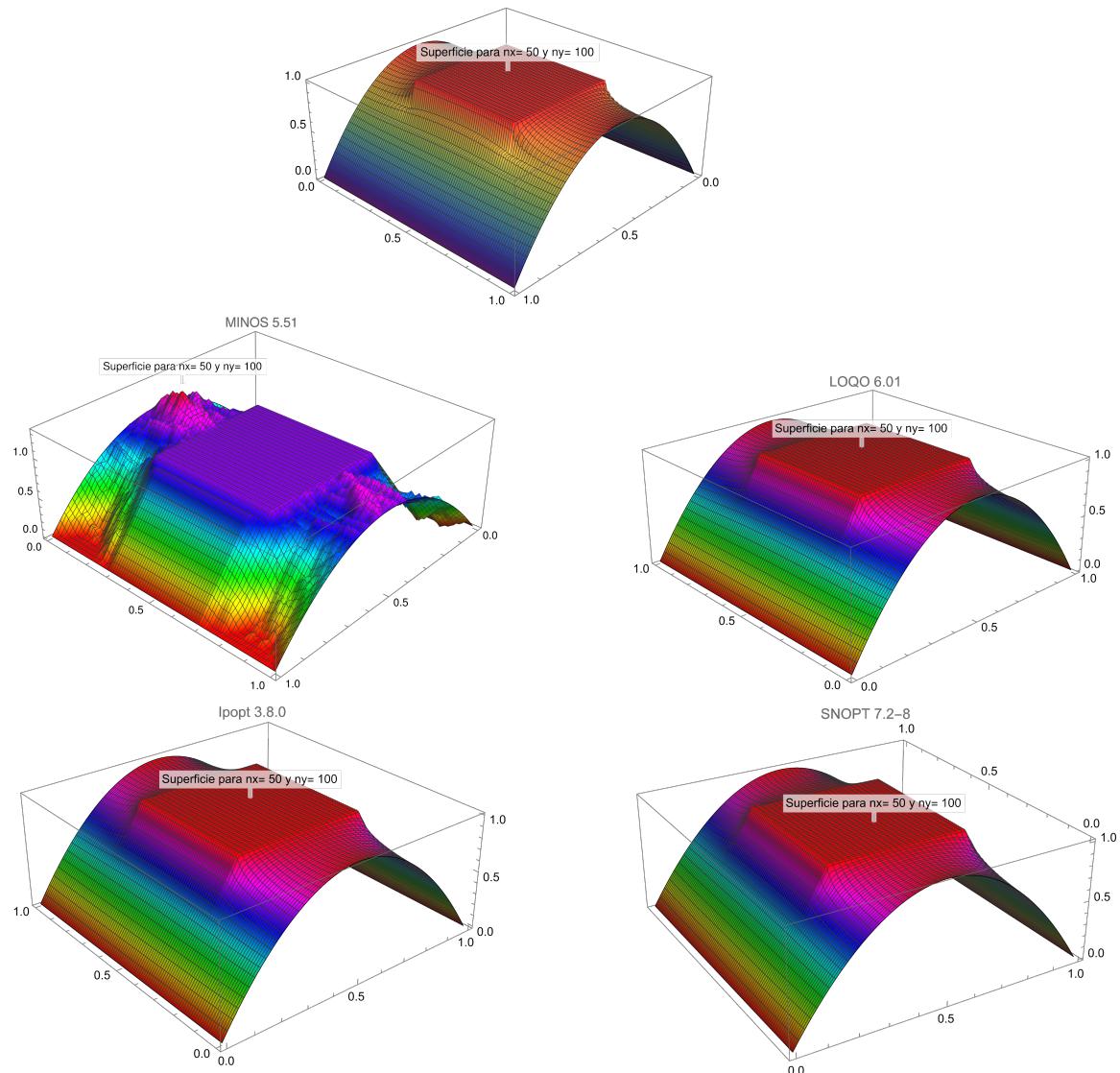
Veamos los resultados obtenidos gráficamente, y se ve que para el *solver* MINOS la gráfica no consigue formar la superficie que andamos buscando. Se genera con muchas irregularidades y no consigue centrar bien el obstáculo como otros *solvers*. En las próximas gráficas, al aumentar el número de puntos de la partición obtenemos el borde del obstáculo más suave, pero no se aprecian grandes cambios entre los *solvers*, excluyendo MINOS.



Para $n_y = 75$, volvemos a obtener resultados parecidos a los anteriores, manteniendo MINOS irregularidades en la gráfica y función objetivo. Así que seguimos optando por LOQO.



Finalmente, la última partición vemos que muchos *solvers* de [1] no llegaron a resolverlo u obtienen tiempos muy altos. Y aunque estos resultados fueron obtenidos en un ordenador con peores características que en el que compilamos, siempre tenemos mejores resultados en el solver LOQO.



Cambiando obstáculos

En esta sección hemos decidido hacer pequeños cambios en el obstáculo, o directamente cambiarlo por completo. Además, mostramos cada ejemplo con el borde modificado. Haremos estas gráficas dándole las restricciones para el borde como apartados anteriores, o definiendo todo el borde a 0.

En todos las implementaciones siguientes hemos tomado $n_x = 70$ y $n_y = 70$.

Cubo sobre el obstáculo

Para este cambio solo ha hecho falta introducir otra restricción (el cubo apilado sobre el ya existente) para obtener el nuevo obstáculo. El primero tiene una altura de 1 y el segundo de 1.5.

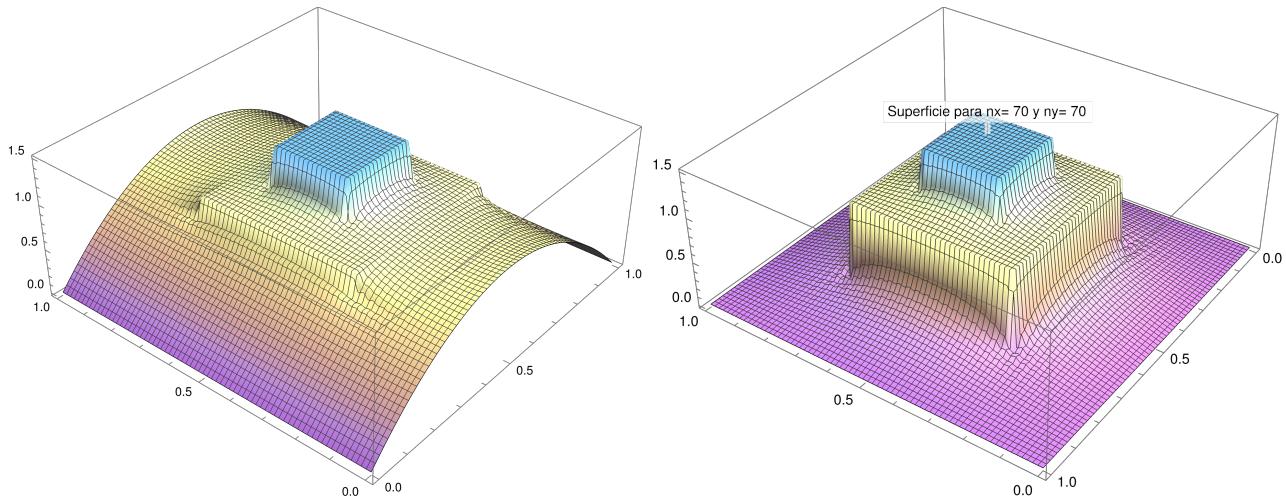
Código: Restricción de ambos cubos

```
R31 = Table[  
    v[i, j] >= 1, {i, Ceiling[nx/4], Floor[(3*nx)/4]},  
    {j, Ceiling[ny/4], Floor[(3*ny)/4]}];  
  
R32 = Table[  
    v[i, j] >= 1.5, {i, Ceiling[(6*nx)/16], Floor[(10*nx)/16]},  
    {j, Ceiling[(6*ny)/16], Floor[(10*ny)/16]}];
```

Y para cambiar la restricción del borde solo hay que igualar a 0 los valores del $v_{i,0}$ y v_{i,n_y} del borde.

Código: Eliminando borde

```
R11 = Flatten[Table[v[i, j] == 0, {i, 0, nx}, {j, {0, ny}}]];
```



Cubos apilados

El proceso anterior podemos repetirlo las veces que queramos, en este caso amontonando 6 cubos de distintos tamaños.

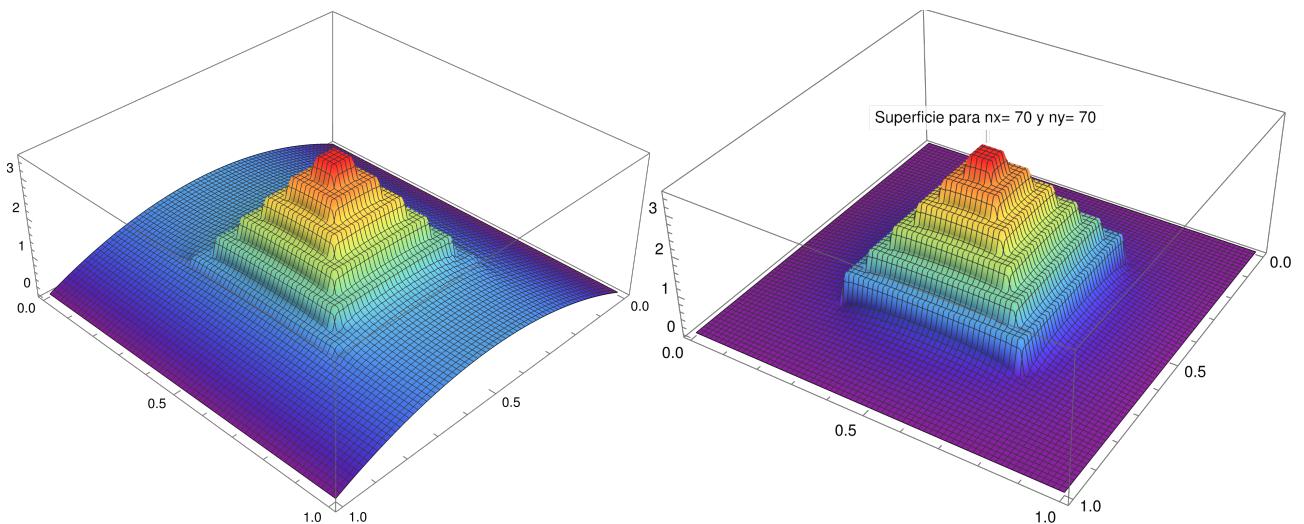
Código: Restricción para los cubos apilados

```
%Número de cubos apilados
ncubs = 6

%Base del primer cubo
base2 = 3/4;
base1 = 1/4;

%Diferencia tamaño cubos
dif = (base2 - base1)/(ncubs*2);
%Altura del primer cubo
tam = 0.3;
%Aumento de altura para cubos sucesivos
alt = 0.5;

R3 = Flatten[
  Table[Table[
    v[i, j] >= (tam + (p)*alt), {i, Ceiling[nx*(base1 + (p - 1)*dif)], 
    Floor[nx*(base2 - (p - 1)*dif)]}, 
    {j, Ceiling[ny*(base1 + (p - 1)*dif)], 
    Floor[ny*(base2 - (p - 1)*dif)]}], {p, 1, ncubs}]];
```

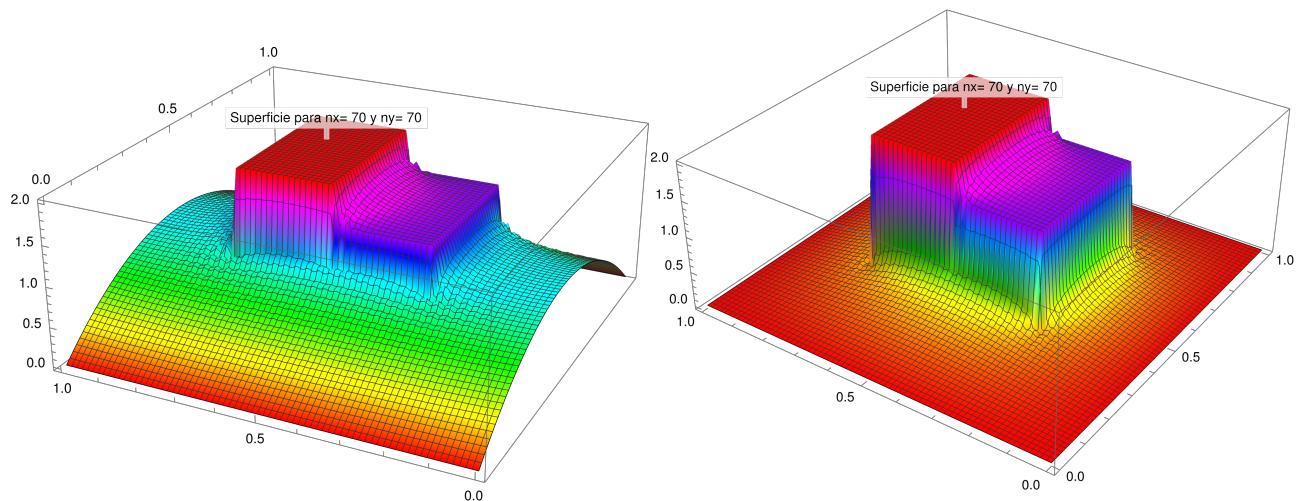


Cubos con distinta altura

En este apartado, en vez de poner un cubo sobre otro, situaremos uno al lado del otro con unas alturas de 1.5 y 2.

Código: Restricciones para los cubos

```
R31 = Table[  
    v[i, j] >= 1.5, {i, Ceiling[nx/3], Floor[(3*nx)/4]},  
    {j, Ceiling[ny/4], Floor[(ny)/2]}];  
  
R32 = Table[  
    v[i, j] >= 2, {i, Ceiling[nx/3], Floor[(3*nx)/4]},  
    {j, Ceiling[ny/2], Floor[(3*ny)/4]}];
```



Cilindro

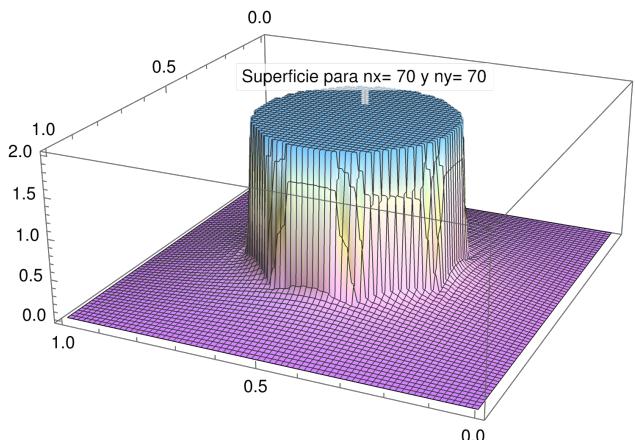
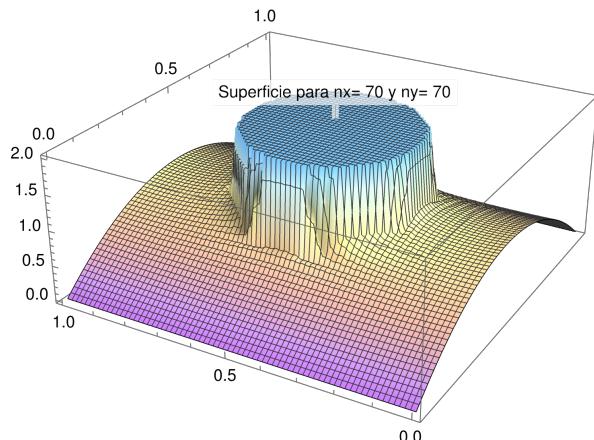
En esta sección cambiamos directamente el obstáculo y decidimos poner un cilindro.

Código: Restricciones para puntos encima del cilindro

```
cilindro = Select[Tuples[{Range[1, nx], Range[1, ny]}],  
Norm[{#[[1]] - nx/2, #[[2]] - ny/2, 0}] <= nx/4 &];  
  
R3 = Table[v[j[[1]], j[[2]]] >= 2, {j, cilindro}];
```

Código: Restricciones para puntos fuera del cilindro

```
R4a7 = Table[v[i, j] >= 0, {i, 1, nx - 1}, {j, 1, ny - 1}];
```



Esfera

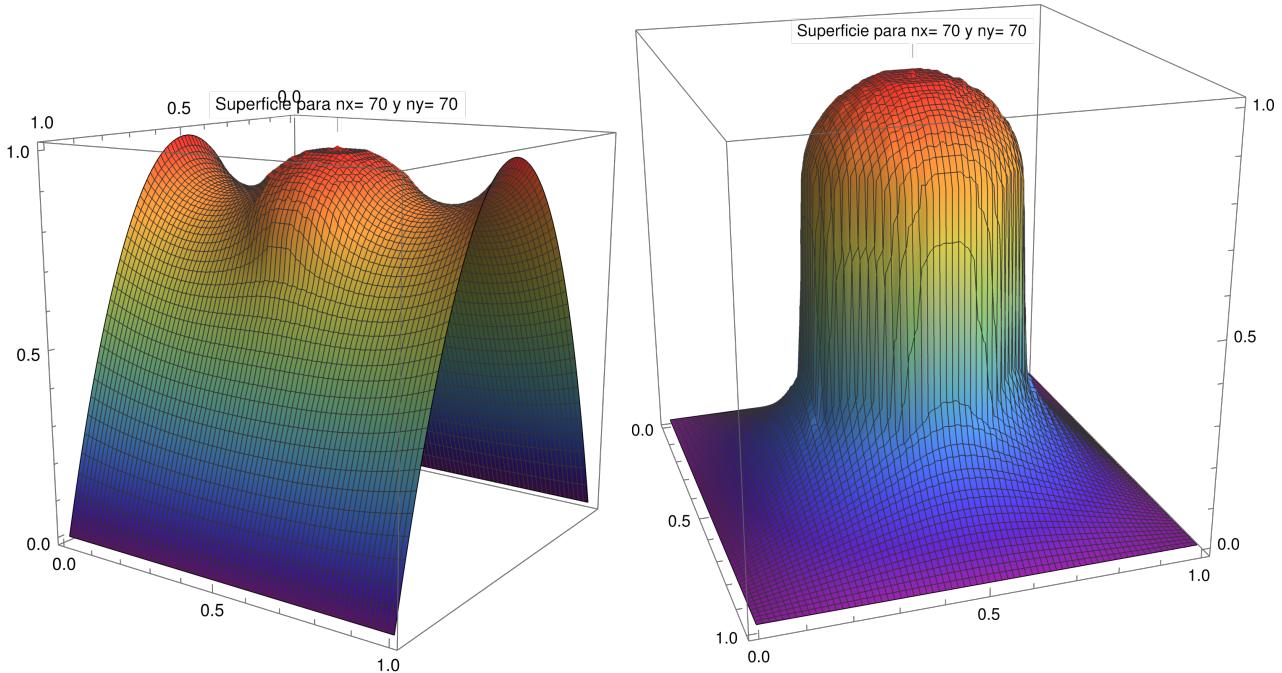
Y por último, cambiamos el obstáculo a una esfera. Esta necesita de muchos puntos para obtener un borde definido y se aprecie mejor.

Código: Restricciones para puntos encima de la esfera

```
esfera = Select[Tuples[{Range[1, nx], Range[1, ny], Range[1, nx]}],  
    Norm[{#[[1]] - nx/2, #[[2]] - ny/2, #[[3]] - 3*nx/4}] <= nx/4 &];  
  
R3 = Table[v[j[[1]], j[[2]]] >= j[[3]]/nx, {j, esfera}];
```

Código: Restricciones para puntos fuera de la esfera

```
R4a7 = Table[v[i, j] >= 0, {i, 1, nx - 1}, {j, 1, ny - 1}];
```



BIBLIOGRAFÍA

- [1] ELIZABETH D. DOLAN, JORGE J. MORE Y TODD S. MUNSON, (2004). *Benchmarking Optimization Software with COPS 3.0*, Mathematics and Computer Science Division, Technical Report ANL/MCS-TM-273, pp. 29–30.
- [2] A. FRIEDMAN, (2000). *Free boundary problems in science and technology*, Notices Amer. Math. Soc., 47, pp. 854–861.
- [3] ÁREA MÍNIMA. FRWIKI. Disponible en: https://es.frwiki.wiki/wiki/Surface_minimale [consulta: 9 mayo 2023].
- [4] PROBLEMA DE PLATEAU. WIKIPEDIA. Disponible en: https://es.wikipedia.org/wiki/Problema_de_Plateau [consulta: 9 mayo 2023].
- [5] RANDALL L. EUBANK , (1999). *Nonparametric Regression and Spline Smoothing*, 2nd Edition, ISBN 9780429182679.
- [6] A. QUARTERONI, R. SACCO Y F. SALERI, (2010). *Numerical Mathematics*, 2nd edition, Springer.
- [7] CITAS Y ELABORACIÓN DE BIBLIOGRAFÍA: EL PLAGIO Y EL USO ÉTICO DE LA INFORMACIÓN: ESTILO PARA CIENCIAS. UNIVERSIDAD AUTÓNOMA DE MADRID. BIBLIOTECA BIBLIOGUÍAS Disponible en: https://biblioguías.uam.es/citar/estilo_ciencias [consulta: 31 marzo 2023].