

Universidad de Cádiz

PNLCC
Catalytic Cracking of Gas Oil

ADRIÁN FERNÁNDEZ TEJADA

ADRIÁN SÁNCHEZ LOUREIRO

2023

INTRODUCCIÓN AL PROBLEMA.....	2	RESULTADOS	9
FORMULACIÓN DEL PROBLEMA	4	IMPLEMENTACIÓN EN AMPL	12
CÓDIGO A DESTACAR	6		



CATALYTIC CRACKING OF GAS OIL

Introducción al problema

Para realizar referenciar los documentos empleados, nos hemos guiado por [5].

Este problema de Programación No Lineal se basa en la reacción del cracking catalítico del gasoil. Tenemos como objetivo determinar los coeficientes de la reacción.

Antes de ver las ecuaciones que determinan esta reacción, ¿Qué ocurre durante el cracking catalítico?.

En general, consiste en la descomposición de una molécula compleja con ayuda de un catalizador en otras más pequeñas. En nuestro caso consiste en romper, por elevación de temperatura con ayuda de catalizadores, las moléculas de ciertos hidrocarburos con el fin de aumentar la proporción de los más útiles.

Un catalizador es una sustancia (compuesto o elemento) capaz de acelerar (catalizador positivo) o retardar (catalizador negativo o inhibidor) una reacción química, permaneciendo este mismo inalterado (no se consume durante la reacción).

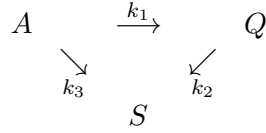
El propósito es craquear hidrocarburos pesados cuyo punto de ebullición es igual o superior a los 315°C , y convertirlos en hidrocarburos livianos de cadena corta cuyo punto de ebullición se encuentra por debajo de los 221°C . Los catalizadores usualmente se componen por óxido de silicio (SiO_2) y alúmina (Al_2O_3). El mineral más comúnmente usado para este fin es la faujasita.

El craqueo hace que la refinería pueda adaptarse a la demanda. Las personas a cargo de la programación de la producción deciden como obtener la destilación a través de los diversos procesos de conversión, para adecuar la cantidad y calidad de los productos finales, según la demanda.

El primer uso comercial del craqueo catalítico se produjo en 1915, utilizando cloruro de aluminio para romper catalíticamente aceites de petróleo pesado.

Este proceso un objetivo puede ser el obtener la máxima concentración de gasolina a partir de gasoil, por razones de interés económico. Sin embargo, siguiendo [4] estudiaremos el comportamiento de la reacción de principio a fin, hasta que se consuman ambos productos (transformándose a otros derivados).

Nuestra reacción se ve representada como.



En él sintetizamos el cracking catalítico de gasoil (A) a gasolina (Q) y otros subproductos (S). El siguiente modelo de la reacción fue propuesto por Froment y Bischoff en [1], descrito por las siguientes ecuaciones:

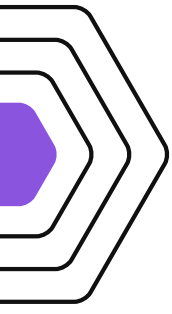
$$\left\{ \begin{array}{l} \frac{dy_A}{dt} = -(k_1 + k_3)y_A^2 \\ \frac{dy_Q}{dt} = k_1y_A^2 - k_2y_Q \end{array} \right. \quad (1.1)$$

Estas ecuaciones modelizan la variabilidad de las proporciones de gasoil (y_A) y gasolina (y_Q). Las constantes cinéticas k_1 , k_2 y k_3 representan las velocidades de reacción de los distintos componentes anteriormente citados.

Observamos que en este modelo, la proporción de gasoil (A) disminuye según las velocidades de reacción k_1 y k_3 , a sus derivados S y Q de manera cuadrática. En cambio, la proporción de gasolina (Q) aumenta según la reacción de gasoil a gasolina y disminuye según la reacción de gasolina a los otros productos S con velocidad k_2 .

Además, los autores en [1] muestran que los parámetros estimados por el modelo son muy próximos a los valores reales.

Las definiciones han sido extraídos de [2].



Formulación del problema

La formulación del problema está inspirada en [4] el cual se guía por [3].

Nuestro problema consiste en, dadas unas medidas de concentración en diferentes instantes, encontrar las velocidades k_1 , k_2 y k_3 que ajusten la reacción a dichas mediciones. Durante la resolución de este, denotaremos como $\theta = \{\theta_1, \theta_2, \theta_3\}$ a las velocidades k_1 , k_2 y k_3 de reacción.

Para ello, la función a minimizar es el error entre las mediciones reales y las deseadas.

$$\sum_{j=1}^{21} \|y(\tau_j; \theta) - z_j\|^2 \quad (1.2)$$

donde:

- $z_j = \{z_A^{(j)}, z_Q^{(j)}\}$ son las concentraciones deseadas de gasoil y gasolina medidas en los instantes de tiempo $0 \leq \tau_1 < \dots < \tau_{21} \leq 1$.
- $y(\tau_j; \theta) = \{y_1^{(j)}, y_2^{(j)}\}$ representa a las concentraciones reales de gasoil (y_1) y gasolina (y_2) en el instante τ_j para los coeficientes $\theta = \{\theta_1, \theta_2, \theta_3\}$.

Usando como restricciones las ecuaciones diferenciales del modelo, antes citado, de Froment y Bischoff en [1] que controlan las concentraciones de cada producto.

$$\begin{cases} \frac{dy_1}{dt} = -(\theta_1 + \theta_3)y_1^2 \\ \frac{dy_2}{dt} = \theta_1 y_1^2 - \theta_2 y_2 \end{cases} \quad (1.3)$$

Para tener un número finito de restricciones vamos a discretizar estas ecuaciones diferenciales usando el método de Euler mejorado explícito (Heun).

Tomaremos como instantes 21 puntos equidistantes en $[0, 1]$. Así tendremos, $21 \cdot 2$ restricciones para los distintos instantes $\tau_{j+1} = \frac{j}{20}$ para $0 \leq j \leq 20$. Por tanto, el paso entre cada instante es $h = \frac{1}{20}$. Destacar que en nuestro código en `Mathematica` tenemos la opción elegir número de instantes.

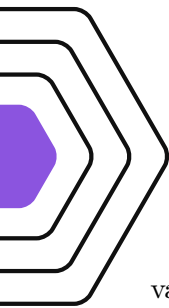
Nuestras restricciones según el método utilizado son:

$$\begin{cases} y_{1,i+1} = y_{1,i} + \frac{h}{2} \cdot \left[f(y_{1,i}, y_{2,i}) + f\left(y_{1,i} + hf(y_{1,i}, y_{2,i}), y_{2,i} + hf(y_{1,i}, y_{2,i})\right) \right] \\ y_{2,i+1} = y_{2,i} + \frac{h}{2} \cdot \left[g(y_{1,i}, y_{2,i}) + g\left(y_{1,i} + hg(y_{1,i}, y_{2,i}), y_{2,i} + hg(y_{1,i}, y_{2,i})\right) \right] \end{cases}$$

con $0 \leq i \leq 20$, siendo $f(y_1, y_2) = -(\theta_1 + \theta_3)y_1^2$ y $g(y_1, y_2) = \theta_1 y_1^2 - \theta_2 y_2$.

En este proceso industrial, partimos de una concentración inicial de solo gasoil. Así pues, nuestros valores iniciales (proporciones iniciales) serán $y_0 = \{y_{1,0}, y_{2,0}\} = \{1, 0\}$.

Obtendremos las mediciones z_j , para $1 \leq j \leq 21$, aplicando una perturbación relativa aleatoria de 0.1 a los valores exactos de cada instante, usando como valores de velocidad de reacción $\theta = \{12, 8, 2\}$.



Código a destacar

Para encontrar solución a nuestro problema, empleamos la función `FindMinium`, que dadas las variables, valor inicial de las anteriores, restricciones discretizadas y función objetivo, devuelve un mínimo local.

Como idea de mejora para obtener unos valores *theta* más parecidos a las mediciones, decidimos añadir puntos intermedios al tiempo. Estos puntos satisfacen las restricciones discretizadas, obteniendo así más puntos para la gráfica de concentraciones.

Hemos definido los 21 instantes como `insts`. En las siguientes líneas de código, vemos como obtenemos los puntos intermedios a partir de estos, según el número `npasos` deseado.

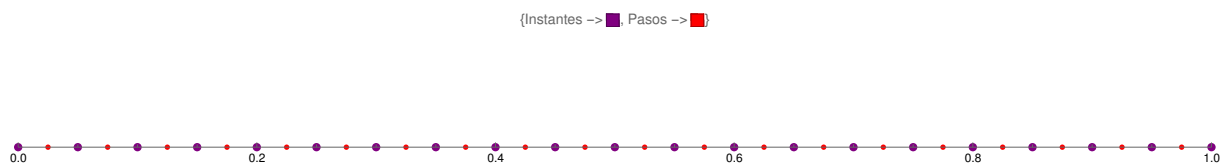
Código: Instantes

```
ninsts = 21;
insts = Table[i*1./(ninsts - 1), {i, 0, ninsts - 1}];

npasos = 2;

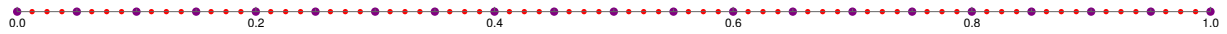
insts2mit = Table[
    Table[
        j*(insts[[i + 1]] - insts[[i]])*1./npasos + insts[[i]],
        {j, 1, Length[insts] - 1}
    ],
    {i, 1, npasos - 1}
]
insts2 = Sort[Join[Flatten[insts2mit], insts]]
```

Gráficamente vemos los siguientes



Y para `npasos = 4`:

{Instantes -> ■, Pasos -> ■}



Hemos definido las variables en `VarP` así:

Código: Variables

```
VarP = Flatten[
  Join[
    Flatten[Table[{y1[i], y2[i]}, {i, 1, Length[insts2] - 1}], 1],
    {\[Theta]1, \[Theta]2, \[Theta]3}
  ]
];
```

Hemos definido la variable `RestriccionP` con nuestras restricciones así:

Código: Restricciones

```
f[y1_, y2_] := -(\[Theta]1 + \[Theta]3) y1*y1;
g[y1_, y2_] := \[Theta]1*y1*y1 - \[Theta]2*y2;

RestriccionesP = Flatten[Join[
  Table[
    y1[i + 1] ==
    y1[i] + (insts2[[i + 2]] -
      insts2[[i + 1]])*(f[y1[i], y2[i]] +
      f[y1[i + 1], y2[i + 1]])/2, {i, 0, Length[insts2] - 2}],
  Table[
    y2[i + 1] ==
    y2[i] + (insts2[[i + 2]] -
      insts2[[i + 1]])*(g[y1[i], y2[i]] +
      g[y1[i + 1], y2[i + 1]])/2, {i, 0, Length[insts2] - 2}],
  {\[Theta]1 >= 0, \[Theta]2 >= 0, \[Theta]3 >= 0}
]];
```

Y por último la función objetivo en `FObjP` como:

Código: Función objetivo

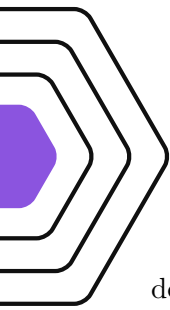
```
FObjP = Sum[(y1[npasos*j] - z1[[j + 1]])^2 + (y2[npasos*j] -  
          z2[[j + 1]])^2 , {j, 0, Length[insts] - 1}];
```

Se puede observar que para las restricciones usamos los tiempos `insts2` y para la función objetivo `insts`.

Finalmente resolvemos con:

Código: FindMinimum

```
FindMinimum[{FObjP, RestriccionesP}, VarP]
```

Resultados

En esta sección, veremos las soluciones obtenidas tomando como mediciones los siguientes valores de z , ya que el error que se le introduce es aleatorio en cada compilación.

```
z1 = {1.‘, 0.6347699836831255‘, 0.388832038042773‘,  
0.2993403752143724‘, 0.2798377320318468‘, 0.21449317633161508‘,  
0.18048680513448762‘, 0.15485515623326518‘, 0.1513021201326371‘,  
0.14257911326636935‘, 0.1315771612269763‘, 0.10441612864972143‘,  
0.1141623204932389‘, 0.10187743023408839‘, 0.09152416536973931‘,  
0.08596399907908388‘, 0.0835082736466175‘, 0.07402553195786021‘,  
0.06933645090147744‘, 0.07044205263873547‘, 0.06941467999811421‘};
```

```
z2 = {0.‘, 0.2908919075792551‘, 0.307701420909982‘,  
0.2936304928086301‘, 0.21685931579500892‘, 0.1717110224067054‘,  
0.14338007708238026‘, 0.10869333150487794‘, 0.09171358018260617‘,  
0.0746851208990134‘, 0.05624565574102057‘, 0.04441887673120921‘,  
0.03578684159544259‘, 0.029132849838211308‘, 0.024418110715885684‘,  
0.01934701038815286‘, 0.015732840098499314‘,  
0.015794546528213684‘, 0.013227477458176946‘,  
0.011862366396319152‘, 0.010050304584443415‘};
```

Resolviendo con `FindMinimun` y empleando la función `RepetedTiming` para medir el tiempo medio empleado, el programa tarda de media 0,058434 s en obtener como valor de la función objetivo 0,00727786 y la solución de los parámetros $\theta = \{10,6272, 7,5945, 2,36133\}$. Podemos ver gráficamente en 1.1 como se aproxima la curva a las mediciones z (puntos azules para el gasoil y puntos naranjas para la gasolina):

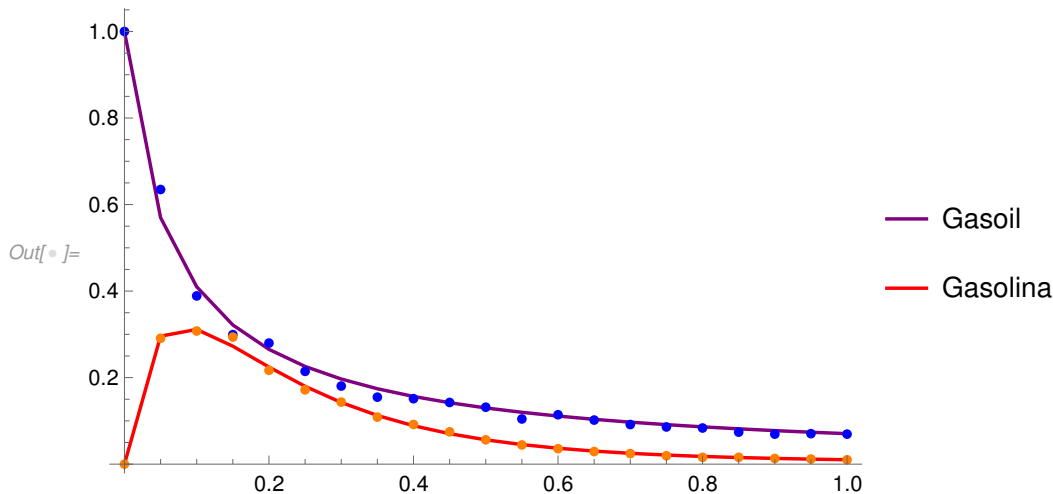


Figura 1.1

La implementación con 4 pasos, tarda de media 0,178424 s en obtener como valor de la función objetivo 0,0051416 y la solución de los parámetros $\theta = \{12,297, 8,06854, 1,69653\}$. Podemos ver gráficamente en 1.2 que la curva es más ajustada a las mediciones z :

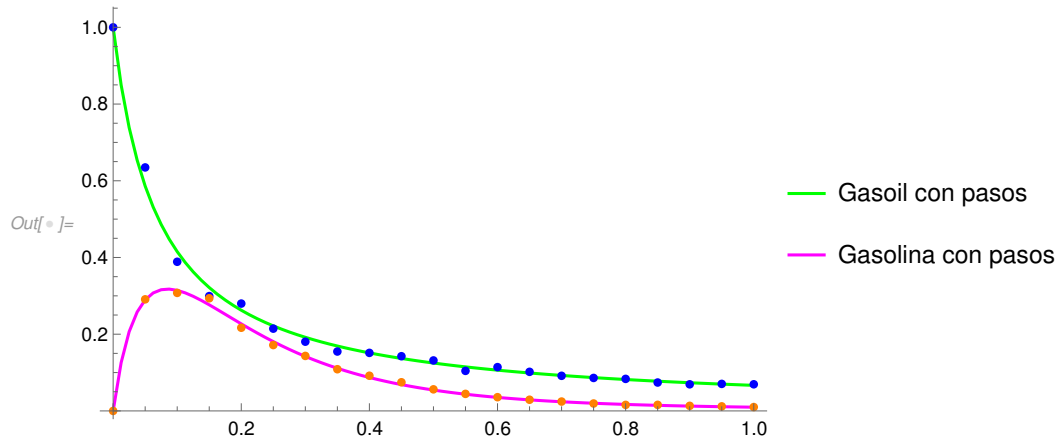


Figura 1.2

Observamos en esta última gráfica 1.3 que las curvas que forman las restricciones sin pasos intermedios son muy poligonales. Por otra parte, tenemos las otras que son más suaves y se ajustan más a las mediciones z .

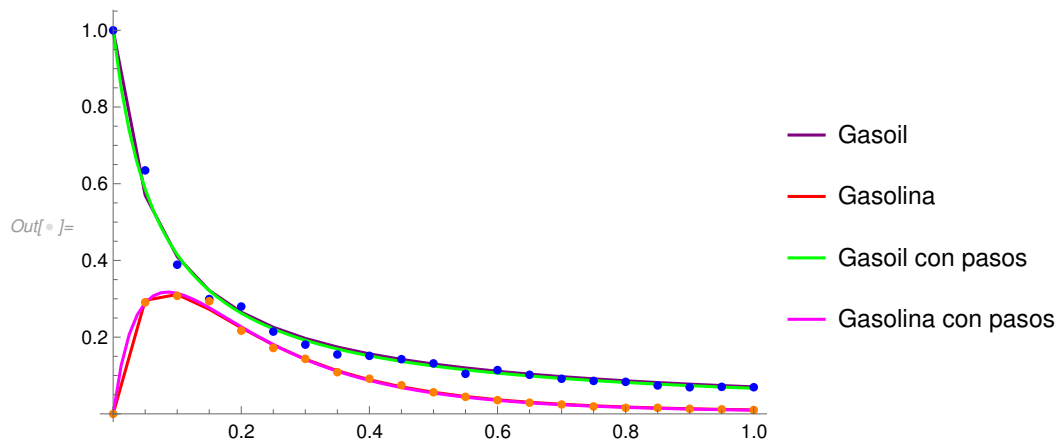


Figura 1.3

Podemos hacer una comparativa también con los valores de la función objetivo y los tiempos empleados. Vemos que las restricciones sin pasos intermedios tarda mucho menos que la otra alternativa, siendo el tiempo de cada una de 0,058434 s y 0,178424 s, respectivamente. Este aumento del tiempo nos da como resultado una mejora en el mínimo de la función objetivo, siendo la primera de 0,00727786

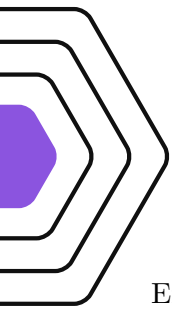
y la implementación de 4 pasos de 0,0051416, y siendo este último mejor que los obtenidos en el Bechmarking [4], en general 0,00523659.

No hacemos una comparativa del tiempo con el Bechmarking [4], ya que de cualquier forma obtenemos mejores tiempos. El menor tiempo empleado en el [4] es de 0,4 s por el software LQO. También hay que ser conscientes de que usamos unos ordenadores con mejores características que en [4], y el programa *Mathematica* ha sido mejorado en muchos aspectos.

Por último, comentar que diferencia obtenemos en los valores de θ según el número de pasos que demos.

- Para 2 pasos obtenemos $\theta = \{11,951, 7,97145, 1,8427\}$.
- Para 3 pasos tenemos los valores $\theta = \{12,2068, 8,04329, 1,73504\}$.
- Para 4 vimos antes que valían $\theta = \{12,297, 8,06854, 1,69653\}$.

Podemos ver que para 2 pasos los coeficientes están muy próximos a los tomados para calcular las mediciones z , $\theta = \{12, 8, 2\}$. Podemos pensar que a mayor número de pasos, obtenemos peores valores al alejarse de $\theta = \{12, 8, 2\}$. Sin embargo, recordar que hemos inducido error a las mediciones z obtenidas a partir de $\theta = \{12, 8, 2\}$. Luego, probablemente, a medida que añadamos pasos, las soluciones obtenidas se adaptaran mejor a las mediciones con error (estas se registrarán por unos valores de θ similares a los originales pero no exactamente esos).



Implementación en AMPL

El problema resuelto en AMPL se encuentra en la carpeta **AMPL** del archivo **12-CRACKING-FTA_SLA.zip**. En esta carpeta, encontramos dos subdirectorios **ESTÁNDAR** y **PASOS**, con, respectivamente, la implementación del problema y del problema añadiendo pasos intermedios.

En esta sección veremos qué resultados obtenemos con la implementación de nuestro problema en AMPL, usando diferentes solver para poder comparar los resultados. Hemos resuelto el problema para los solver **MINOS**, **LOQO**, **CONOPT** y **SNOPT**, y haremos una tabla comparativa con los datos obtenidos en el Bechmarking [4].

Solver	Bechmarking	Problema estándar	Implementación con 4 pasos
MINOS	1.85 s	0.036793 s	0.030263 s
Función objetivo	0.00523659	0.00727788	0.00514162
θ_1		10.62717913	12.29705131
θ_2		7.59450775	8.06855859
θ_3		2.36132520	1.69650776
LOQO	0.4 s	0.054354 s	0.050677 s
Función objetivo	0.00523659	0.00727784	0.00514099
θ_1		10.62716660	12.29693822
θ_2		7.59449957	8.06849384
θ_3		2.36132847	1.69650342
CONOPT		0.0720609999 s	0.070346 s
Función objetivo		0.00727788 (Locally)	0.00514162 (Locally)
θ_1		10.627174994	12.29705129
θ_2		7.594503410	8.06855858
θ_3		2.361329097	1.69650778
SNOPT	2.41 s	0.087982 s	0.089056 s
Función objetivo	0.00523659	0.00727788	0.00514162
θ_1		10.627174994	12.29705129
θ_2		7.594503410	8.06855858
θ_3		2.361329097	1.69650778
Mathematica		0,058434 s	0,178424 s
Función objetivo		0,00727786	0,0051416
θ_1		10,6272	12,297
θ_2		7,5945	8,06854
θ_3		2,36133	1,69653

Cuadro 1.1: Resultados obtenidos en diferentes implementaciones

El Bechmarking [4] 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 7.03**, **CONOPT 3.17A**, **SNOPT 7.5-1.2** y **Mathematica 13** (véase 1).

Destacar que tanto las versiones empleadas como el ordenador por el Benchmarking [4] 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.

Además, en Benchmarking [4] han usado el método de colocación para la resolución de las ecuaciones diferencias. En este documento, empleamos el método de Euler mejorado explícito. Como empleamos métodos diferentes, tomamos la columna que con los datos $n_h = 100$. Podemos ver que como cabía esperar tenemos una mejora en el tiempo de compilación, pero en la implementación del problema estándar no hay una mejora de la función objetivo. Así que por un pequeño aumento de tiempo podríamos usar la implementación de 4 pasos y obtenemos una mejor función objetivo que en Benchmarking [4].

Observamos gráficamente los valores obtenidos para el problema estándar en figura 1.1 y para la implementación con pasos en figura 1.2 al obtener valores con diferencias despreciables de las velocidades de reacción θ en general.

Ahora, haremos una comparación solo de las versiones actuales. Obteniendo el mejor resultado del problema estándar con el solver MINOS, con un tiempo de 0.036 s. Las funciones objetivo no sufren gran cambio y las velocidades de reacción θ son prácticamente iguales. En la implementación de 4 pasos se sigue obteniendo el mejor resultado con MINOS con un tiempo de 0.030 s, y las funciones objetivo son muy próximas. En cambio, es el único solver que mejora el tiempo usando una implementación de 4 pasos y también se obtiene un menor valor de la función objetivo. Por tanto, la mejor opción será el solver MINOS para la implementación con 4 pasos, ya que obtenemos el menor tiempo y la mejor función objetivo.

Por último, graficaremos los valores de gasoil y gasolina que obtenemos con estos solvers. Tomaremos solo las soluciones de MINOS, ya que llegamos a la conclusión de que era la mejor opción. Pero con todos los solvers obtenemos resultados equivalentes.

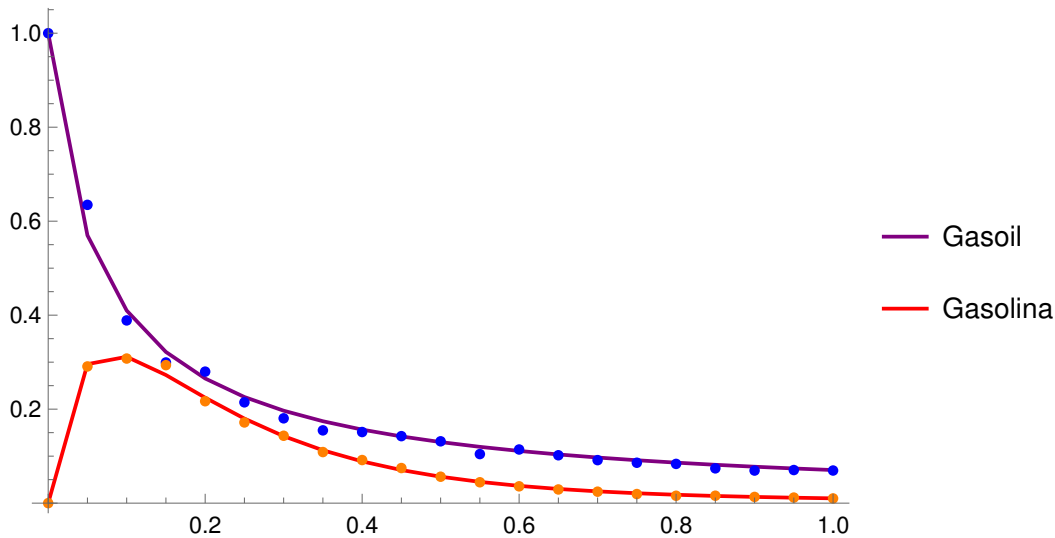


Figura 1.4

Por otro lado, en 1.5 vemos las gráfica del problema con pasos.

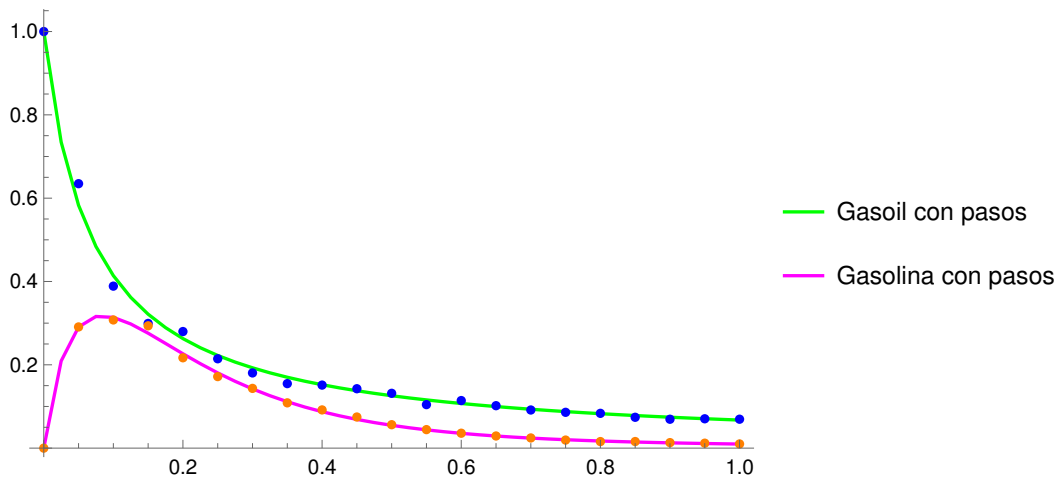


Figura 1.5

BIBLIOGRAFÍA

- [1] GILBERT F. FROMENT, KENNETH B. BISCHOFF Y JURAY DE WILDE, (1979). *Chemical Reactor Analysis and Design*, 1^a ed. New York Wiley & Sons, Incorporated, John, 1979.
- [2] *FLUID CATALYTIC CRACKING*. WIKIPEDIA. Disponible en: https://en.wikipedia.org/wiki/Fluid_catalytic_cracking [consulta: 31 marzo 2023].
- [3] I.-B. TJOA AND L. T. BIEGLER, (1991). *Simultaneous solution and optimization strategies for parameter estimation of differential-algebraic equations systems*, Ind. Eng. Chem. Res., 30 (1991), pp. 376–385.
- [4] 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.
- [5] *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://biblioguias.uam.es/citar/estilo_ciencias [consulta: 31 marzo 2023].