

Desarrollo de una herramienta para la simulación de reactores



Grado en Ingeniería Química

Trabajo Fin de Grado

Autor:

Isabela Fons Moreno-Palancas

Tutor/es:

Rubén Ruiz Femenia

Junio 2020



Universitat d'Alacant
Universidad de Alicante

ÍNDICE

1	Resumen.....	4
2	Palabras clave.....	4
3	Introducción	5
4	Antecedentes	7
5	Objetivos	8
6	Fundamento teórico.....	9
6.1	Cálculo de la velocidad de reacción.....	9
6.2	Simulación de un reactor de flujo pistón.....	11
6.3	Simulación de un reactor continuo de tanque agitado.....	12
6.4	Simulación de un reactor discontinuo de tanque agitado.....	13
6.5	Bypass y recirculación	14
6.6	Asociación de reactores.....	16
6.7	Optimización del reactor	17
6.8	Camino de temperatura óptima	19
6.9	Estimación económica	22
6.9.1	Coste del capital (CAPEX).....	22
6.9.2	Coste operacional (OPEX)	24
6.9.3	Coste total anualizado (TAC).....	24
7	Metodología	25
7.1	Sistema reaccionante.....	26
7.1.1	Cálculo de la velocidad de reacción	27
7.1.2	Comunicación con Aspen HYSYS.....	28
7.2	Reactor	31
7.3	Corriente.....	34
7.4	Diseño de la interfaz gráfica	34

8	Resultados	36
8.1	Ejemplo de resolución.....	47
9	Discusión de resultados.....	53
10	Conclusión	60
11	Referencias.....	62
12	Anexo I: Ejemplos de uso de ReactorApp a partir de código	63
12.1	Simulación de un reactor tubular de lecho fijo	63
12.2	Optimización de un RDTA	65
13	Anexo II: Archivos.....	68

1 RESUMEN

Hoy en día la transformación digital es un proceso que no solo despierta interés en el sector empresarial, sino que ha llegado para quedarse en el ámbito de la educación superior, donde la integración de nuevos recursos tecnológicos en el aprendizaje de la ingeniería química resulta interesante. En este trabajo se ha desarrollado una aplicación capaz de simular el comportamiento de reactores ideales, tanto reactores discontinuos de tanque agitado como RCTA o PFR, realizar una estimación económica preliminar de los diferentes equipos, resolver problemas de optimización con el objetivo de minimizar los costes o el tiempo de residencia y establecer comunicación con otros programas para importar datos. Esta herramienta se fundamenta en un conjunto de funciones implementadas en el lenguaje de MATLAB con las que el usuario puede interactuar de manera directa desde el entorno de trabajo de este programa o bien de una forma más guiada por medio de una interfaz gráfica o *toolbox* diseñada con AppDesigner para resolver problemas de reactores químicos, permitiendo así profundizar en el conocimiento de estos sistemas con un enfoque más interactivo y autónomo.

2 PALABRAS CLAVE

MATLAB Toolbox

E-learning

Reactores químicos

Simulación y optimización de reactores

Aspen HYSYS

3 INTRODUCCIÓN

La aplicación de la conservación de los balances de materia, energía y cantidad de movimiento es uno de los pilares fundamentales de la Ingeniería Química, y en particular del estudio del comportamiento de los reactores químicos. El nivel de detalle que puede llegar a incluir estas expresiones requiere que en muchas ocasiones el ingeniero recurra a software específico de simulación, principalmente para la determinación de las propiedades termodinámicas. MATLAB es un programa de cálculo numérico con la suficiente potencia y flexibilidad para abordar este tipo de problemas, lo cual lo convierte en una de las herramientas más utilizadas en los estudios de ingeniería, y en especial, por los estudiantes del grado en Ingeniería Química para desarrollar tanto sus habilidades de programación como para llevar a cabo tareas de investigación. En este punto, la idea de desarrollar y poner a disposición de otros alumnos del grado una aplicación que funcione como un simulador de reactores químicos en el entorno de MATLAB resulta muy interesante debido a que puede convertirse en un complemento y un apoyo para el estudio del comportamiento de los diferentes tipos de reactores.

Ahora que el sistema educativo afronta una etapa de cambio y transición hacia un modelo de educación que combine la enseñanza presencial con la docencia en línea, una de las utilidades más inmediatas que puede atribuirse a una aplicación de esta índole es la comprobación de los ejercicios propuestos desde diferentes asignaturas, ofreciendo al alumnado la capacidad de trabajar de forma más autónoma, detectando y analizando los posibles errores cometidos. Sin embargo, sus posibilidades pueden ir más lejos ya que, además de obtener los resultados finales del problema planteado, también pueden generarse diferentes representaciones para visualizar el comportamiento del sistema, lo cual puede conducir a una mejor comprensión de la materia. Por otra parte, puede resultar muy útil de cara a la implantación de laboratorios virtuales, cuya necesidad ha quedado patente ante escenarios como el que estamos viviendo en este año 2020 por una la pandemia a nivel mundial. Así, los alumnos podrían desarrollar sus prácticas a distancia a través de un entorno de realidad virtual, que respondería a las acciones de los alumnos sobre los equipos gracias a los cálculos que describen el comportamiento del sistema de reactores, y que estarían basados en herramientas análogas a las que se presenta en esta memoria. Para las prácticas presenciales, este laboratorio virtual también sería beneficioso, puesto que los alumnos lo podrían utilizar durante la preparación de las mismas para tener una idea preliminar de cómo debería desarrollarse el experimento o cuál puede ser el efecto de las variables que se vayan a modificar, o bien a posteriori, para comparar los datos experimentales con los datos teóricos o para profundizar más sobre la práctica realizada simulando diferentes escenarios.

En definitiva, al tratarse de una herramienta de fácil manejo y que no requiere de grandes conocimientos previos para ser explotada, puede ser de gran ayuda para que los alumnos puedan resolver problemas no solo de simulación, sino también algunas cuestiones relativas a la optimización de reactores, de forma rápida y sencilla. No obstante, cabe destacar que en ningún momento se busca competir con otros paquetes de software profesionales, sino que simplemente se ha pretendido crear una herramienta docente versátil y accesible a un público que busca aprender y consolidar sus conocimientos en el campo de la Ingeniería Química y, más concretamente, en el diseño, simulación y optimización de reactores químicos.

4 ANTECEDENTES

El modelado computacional es un procedimiento fuertemente consolidado en el análisis de procesos químicos. Además, el gran desarrollo tecnológico de los últimos años y la integración de estas nuevas tecnologías como parte fundamental de la educación, ha despertado el interés por la creación de aplicaciones o *toolboxes* empleadas en el campo de la Ingeniería Química tanto a nivel educativo como de investigación.

La mayoría de las aplicaciones desarrolladas hasta el momento comparten un fin común: fomentar la educación a distancia, de forma virtual y autónoma. Uno de los ejemplos más destacados que se encuentran en la bibliografía se trata de una aplicación generada con MATLAB que, a través de una interfaz gráfica, permite recrear una experiencia de laboratorio, hasta el punto de ofrecer al estudiante la posibilidad de modificar diferentes parámetros y ver la evolución del sistema de forma remota (Nieva, 2016). En la propia Universidad de Alicante, los estudiantes de Ingeniería Robótica ya trabajan con RobUAlab (AUROVA, 2008), una aplicación que simula de forma virtual un robot industrial para practicar los comandos básicos de control de un brazo robótico, los cuales se envían a un robot real localizado en el laboratorio AUROVA y los resultados se pueden ver a través de internet. Por tanto, se trata de una línea de trabajo o investigación que, de forma más o menos modesta, ha sido el objeto de numerosos proyectos aunque la mayoría de ellos están enfocados en un ámbito concreto o en la resolución de un problema determinado (Cristea et al., 2005).

Aunque esta falta de versatilidad pueda parecer una limitación, la potencia de este tipo de recursos no pasa desapercibida a nivel empresarial pues incluso existen evidencias de aplicaciones que también se han desarrollado en el entorno de MATLAB con aplicación en el ámbito farmacéutico, donde se utilizan para llevar a cabo simulaciones de diferentes tipos de reactores y aprovechar esta información para la escalada de procesos de síntesis farmacológica (Plouffe et al., 2014).

En vista a lo expuesto anteriormente, una de las líneas de trabajo abiertas está en la creación de herramientas que admitan modificar el sistema reaccionante que se quiera estudiar y probar diferentes tipos de reactores desde una interfaz gráfica de MATLAB, partiendo de unos conocimientos básicos y sin necesidad de instalar ni recurrir otro tipo de simuladores específicos.

5 OBJETIVOS

El objetivo principal del trabajo se basa en el desarrollo de un aplicación informática que simule el comportamiento de diferentes tipos de reactores y optimice aspectos relacionados con su diseño. Para que sea un recurso útil y eficaz, dicha aplicación se espera que cumpla con una serie de características básicas:

- Rigor y exactitud: los resultados ofrecidos por la aplicación deben ser totalmente fiables.
- Versatilidad: capacidad de adaptarse para resolver problemas de muy distinta índole.
- Claridad: se busca que el manejo de la aplicación sea lo más sencillo e intuitivo posible.

Para poder llevar a cabo el objetivo principal, la realización del presente trabajo puede subdividirse en una serie de etapas.

1. Definición de los modelos matemáticos que describen el comportamiento de los diferentes reactores y sistemas considerados, planteamiento de los problemas a resolver y de la estimación de costes. Conviene destacar que estas tareas conllevan una etapa previa de documentación y recopilación de información bibliográfica para escoger qué ecuaciones y métodos se deben emplear.
2. Desarrollo del código en el lenguaje de MATLAB que ejecute todas las operaciones necesarias. Con la idea de que esta herramienta pueda ser fácilmente actualizable, e incluso permita incorporar nuevas funcionalidades, se ha intentado, en la medida de lo posible, una implementación que siga los principios de la programación orientada a objetos (*“Oriented Object Programming”*).
3. Creación de la interfaz gráfica desde el entorno de App Designer para que el usuario sea capaz de interactuar con la aplicación de la forma más intuitiva posible.
4. Etapa de validación que permita corregir y depurar los fallos cometidos, así como adquirir un mejor conocimiento de cuáles son las limitaciones y las líneas de mejora que presenta la aplicación desarrollada.

6 FUNDAMENTO TEÓRICO

En esta sección se presentan los balances de conservación de materia, energía y cantidad de movimiento que se han implementado en [ReactorApp](#) para simular cada uno de los tipos de reactor estudiado, así como aquellas ecuaciones en las que se basan los cálculos fundamentales realizados por el simulador que se ha programado en MATLAB, como es el caso de la velocidad de reacción. Finalmente, se enuncia de forma general el planteamiento seguido en los problemas de optimización.

6.1 Cálculo de la velocidad de reacción

Para calcular la velocidad de reacción se ha escogido el modelo de Langmuir–Hinshelwood–Hougen–Watson (LHHW), ampliamente utilizado por programas profesionales como Aspen Plus y Aspen HYSYS en sus simulaciones. Este modelo describe el comportamiento de reacciones que se llevan a cabo en presencia de un catalizador heterogéneo (Moulijn et al., 1993), cuyo mecanismo de reacción básico consta de tres etapas:

1. Adsorción de los reactivos sobre la superficie del catalizador
2. Reacción química entre los reactivos
3. Desorción de los productos hacia el seno de la fase gaseosa

A continuación, se muestra la expresión general del modelo de LHHW.

$$r = \frac{\text{factor cinético} \cdot \text{fuerza impulsora}}{\text{término de adsorción}} \quad (1)$$

El factor cinético siempre contiene la constante cinética de la etapa que controla la reacción y puede evaluarse recurriendo a la ecuación de Arrhenius.

$$\text{factor cinético} = k_i = k_{0,i} \exp\left(-\frac{E_{a,i}}{RT}\right) \quad (2)$$

Por otra parte, la fuerza impulsora es proporcional a la concentración de reactivos y este término no contiene ningún parámetro relacionado con la presencia de un catalizador. En la aplicación desarrollada, si no se especifica nada respecto del orden parcial al que está elevada la concentración de cada reactivo, se considera que las reacciones son elementales, es decir, el orden de reacción coincide con el valor absoluto del coeficiente estequiométrico de los reactivos que participan en la reacción. En general, la fuerza impulsora puede expresarse como un productorio de la concentración de las distintas especies, donde N es el número de componentes y ν el orden parcial del componente j en la reacción i .

$$fuerza\ impulsora = \prod_{j=1}^N C_j^{v_j} \quad (3)$$

El término de adsorción que aparece en el denominador simboliza la reducción de sitios activos en la superficie del catalizador, que depende de la interacción de las diferentes sustancias participantes en la reacción con el sólido. Por tanto, su complejidad será diferente según el mecanismo considerado. En este trabajo, se ha utilizado la expresión que describe el caso general, en el que la velocidad de adsorción o desorción de un compuesto es proporcional a la concentración o presión parcial del mismo y cuya constante de proporcionalidad o constante cinética de adsorción (Perry, 1997).

$$término\ de\ adsorción = 1 + \sum_1^M k' \prod_{j=1}^N C_j^{\beta_j} \quad (4)$$

La constante cinética de adsorción (k') depende de la temperatura al igual que ocurre con la constante de velocidad de reacción por lo que su valor puede calcularse empleando también el modelo de Arrhenius y el parámetro β_j es exponente al que se eleve la concentración del compuesto adsorbido sobre el catalizador.

A la vista de la ecuación 4, en caso de que la reacción no se lleve a cabo en presencia de un catalizador o simplemente no se disponga de información suficiente sobre los parámetros cinéticos del sistema reaccionante con el que se desea trabajar, no será necesario proporcionar ninguno de los valores de k' o β_j y el denominador será igual a la unidad.

A modo de conclusión, cabe destacar la versatilidad que ofrece la elección del modelo de LHHW para el cálculo de la velocidad de reacción ya que resulta apropiado tanto para sistemas que sigan una cinética simple como para otros complejos o no convencionales en los que tiene lugar una reacción catalizada por sólidos.

Por último, cuando se trate de reacciones catalíticas y la velocidad de reacción esté expresada en base a la masa del catalizador, se ha incluido un factor de conversión para tener en cuenta la equivalencia de masa del catalizador a volumen de reactor. Dicho factor está relacionado con la densidad aparente del catalizador.

$$r \left(\frac{[mol]}{[L]^3 \cdot [t]} \right) = (1 - \varepsilon) \rho_s r \left(\frac{[mol]}{[M] \cdot [t]} \right) \quad (5)$$

En la expresión anterior ε es porosidad del lecho (m^3 huecos/ m^3 reactor) y ρ_s la densidad del sólido (kg catalizador/ m^3 catalizador).

6.2 Simulación de un reactor de flujo pistón

La ecuación 6 es la expresión general del balance de materia para este tipo de reactor donde n_j es el flujo molar del componente j , dL es un elemento diferencial a lo largo del eje longitudinal del reactor, S la sección transversal del reactor y r_j la velocidad de producción o consumo de la especie j .

$$\frac{dn_j}{dL} = Sr_j, \quad \forall j \quad (6)$$

Un reactor de flujo pistón en la realidad se asemeja a un intercambiador de calor de tubos en el sentido de que está compuesto por un conjunto de conducciones, generalmente de sección cilíndrica, dispuestas en paralelo. Esta configuración permite un mejor control de la temperatura cuando se trabaja con reactores de gran diámetro. Por este motivo, la sección transversal del reactor será la correspondiente al producto de la sección de un tubo por el número de tubos que formen el reactor.

El balance de energía para este tipo de reactores queda definido por la ecuación 7.

$$\frac{dT}{dL} = \frac{dQ/dL - A \sum_i^R (r_i \Delta H_i)}{\sum_j^N (n_j C_{p,j})} \quad (7)$$

En este caso A no es la sección transversal sino la superficie externa del cilindro a través de la cual tiene lugar el intercambio de calor con el entorno. Además, ΔH_i simboliza la entalpía de la reacción i y $C_{p,j}$ la capacidad calorífica de cada especie j .

Si el reactor es adiabático, el término dQ/dL será nulo, pero si se hace uso de servicios fríos o calientes para controlar la temperatura del reactor, el flujo de calor se evalúa con la siguiente expresión:

$$\frac{dQ}{dL} = 2\pi R U (T - T_w) \quad (8)$$

donde R es el radio de un tubo del reactor, Q es el flujo de calor, U el coeficiente global de transmisión de calor y T y T_w son la temperatura en el interior del reactor y del fluido refrigerante o calefactor, respectivamente.

El uso de esta ecuación lleva implícito realizar la suposición de que la temperatura del fluido calefactor o refrigerante se mantiene constante. Si se realizara un tratamiento riguroso, habría que considerar la variación de la temperatura en la corriente de servicios fríos o calientes. Sin embargo, con el objetivo de simplificar los cálculos se asume como válida la suposición planteada.

Para evaluar cómo varía la presión a lo largo del reactor, cuando se trate de un reactor de flujo pistón no catalítico, es decir, que este constituido únicamente por una o una serie de conducciones cilíndricas, la expresión del balance de cantidad de movimiento es la que se muestra a continuación.

$$\frac{dP}{dL} = \frac{\frac{2fQ_v}{D} + \frac{R}{\alpha P} \left[\sum_j n_j \frac{dT}{dL} + T \sum_j \frac{dn_j}{dL} \right]}{\frac{-S^2}{\sum_j n_j M_j} + \frac{RT}{\alpha P^2} \sum_j n_j} \quad (9)$$

En el balance anterior Q_v es el caudal volumétrico de fluido, D es el diámetro de reactor, P es la presión, R es la constante de los gases ideales y M_j es el peso molecular de la especie j . El parámetro α toma el valor de 0,5 para flujo laminar o 0,96 para flujo turbulento. Finalmente, el factor de fricción se simboliza con la letra f en la ecuación 9. Aunque existe una multitud de expresiones en la bibliografía para calcularlo, se ha elegido la relación de McAdams.

$$f = 0,046 Re^{-0,2} \quad (10)$$

La constante R ha sido definida en unidades del sistema internacional, es decir, con el valor de 8,314 J/mol/K por lo que trabajar en unidades del sistema internacional es de vital importancia cuando se desee tener en cuenta el efecto de la variación de la presión.

Por otro lado, cuando se trata de reactores tubulares de lecho empacado, se prefiere emplear la ecuación de Ergun en lugar de la ecuación 9 para evaluar la evolución de la presión a lo largo del reactor (Fogler, 2006).

$$\frac{dP}{dL} = \frac{-G}{\rho d_p} \frac{(1 - \varepsilon)}{\varepsilon^3} \left[150 \frac{(1 - \varepsilon)\mu}{d_p} + 1,75 G \right] \quad (11)$$

En la ecuación 11, el diámetro de partícula se representa como d_p , ρ y μ hacen referencia a la densidad y viscosidad del fluido, respectivamente, y G es la velocidad másica, calculada como el producto de la densidad del fluido por la velocidad lineal del mismo en el interior de la tubería, por lo que tiene unidades de masa/superficie/tiempo.

6.3 Simulación de un reactor continuo de tanque agitado

En la simulación de reactores tipo RCTA conviene tener en cuenta que se opera en régimen estacionario, es decir, el comportamiento dinámico de un RCTA durante los procesos de parada y puesta en marcha o como respuesta ante una perturbación están fuera del alcance de las posibilidades de este proyecto. Por consiguiente, los balances que se han implementado en el código de MATLAB son:

- Balance de materia

$$\frac{n_j^{entrada} - n_j}{V} = r_j, \quad \forall j \quad (12)$$

- Balance de energía

$$\sum_{j=1}^N \left(n_j^{entrada} \int_{T^{entrada}}^T C_{p,j} dT \right) + V \sum_{i=1}^R r_i \Delta H_i - Q = 0 \quad (13)$$

En el balance de energía, el flujo de transferencia de calor se evalúa utilizando la siguiente expresión:

$$Q = UA (T - T_w) \quad (14)$$

donde A es la superficie de intercambio de calor y el resto de las variables, al igual que se introdujo con anterioridad, hacen referencia al flujo de calor (Q), al coeficiente global de transmisión de calor (U), a la temperatura en el interior del reactor (T) y a la del fluido frío o caliente (T_w). Al igual que en balance de energía para el RFP, también se asume que la T_w toma un valor constante.

A modo de aclaración, conviene mencionar que se ha despreciado la pérdida de presión por rozamiento, asumiendo que la presión en las corrientes de alimento y salida son idénticas para este tipo de reactores.

6.4 Simulación de un reactor discontinuo de tanque agitado

Para el estudio de operaciones que se llevan a cabo por cargas o de forma intermitente, se han implementado los balances de materia y energía asociados a un reactor discontinuo de tipo tanque agitado, cuyas expresiones quedan definidas en las ecuaciones 15 y 16.

$$\frac{dN_j}{dt} = V r_j, \quad \forall j \quad (15)$$

$$\frac{dT}{dt} = \frac{Q - V \sum_i r_i \Delta H_i}{\sum_j N_j C_{p,j}} \quad (16)$$

El término Q, referente al flujo de calor intercambiado entre el reactor y el entorno, se calcula a partir de la misma expresión indicada en el apartado anterior para el RCTA.

En procesos que se lleven a cabo en fase gaseosa, para conocer la relación entre las variables de estado, presión, temperatura y volumen, se utilizará la ley de los gases ideales.

$$pV = \frac{N_{\text{totales}} RT}{V} \quad (17)$$

6.5 Bypass y recirculación

Cuando se pretenda estudiar el comportamiento de un reactor incluyendo una corriente de bypass o de recirculación, los cálculos se basan en los balances correspondientes en el entorno de un mezclador o de un divisor de corrientes, según corresponda. Las expresiones que se han utilizado para dichos balances son válidas tanto para fase gas como fase líquida ya que están desarrolladas en base a caudales molares y no volumétricos.

En la simulación de un reactor con una corriente de bypass, para calcular los moles alimentados al reactor y los moles totales a la salida, es necesario conocer el ratio o relación de bypass, designado como B . Las ecuaciones correspondientes se han deducido atendiendo a la nomenclatura utilizada en la figura 1.

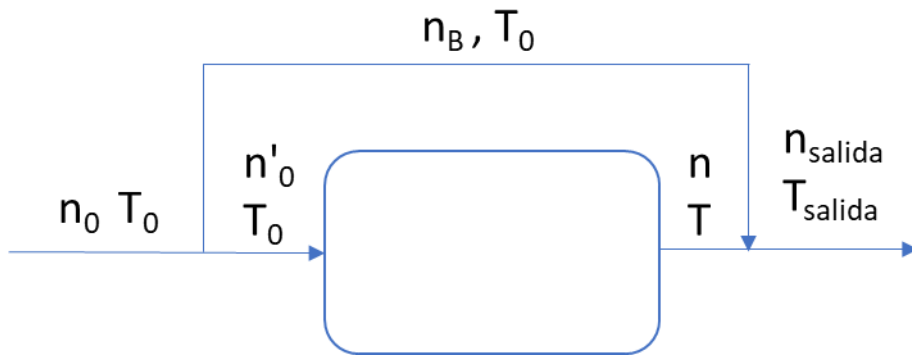


Figura 1: Esquema genérico de un reactor con una corriente de bypass. Fuente: elaboración propia.

$$B = n_B/n'_0$$

$$n_0 = n'_0 + n_B = n'_0 + B n'_0$$

$$n'_0 = \frac{n_0}{1 + B} \quad (18)$$

$$n_{\text{salida}} = n + n_B = n + B n'_0 \quad (19)$$

Además, se ha tenido en cuenta que la corriente de bypass y la de salida del reactor pueden estar a diferente temperatura por lo que la temperatura de la corriente total de producto vendrá determinada a partir del balance de energía en el mezclador (ecuación 20).

$$T_{\text{salida}} = \frac{\sum_j n_j C_{p,j} T + \sum_j n_{B,j} C_{p,j} T_0}{\sum_j (n_j + n_{B,j}) C_{p,j}} \quad (20)$$

Para reactores que cuentan con un sistema de recirculación, al igual que ocurre en los reactores biológicos de las depuradoras que emplean métodos de fangos activos, el procedimiento para deducir las ecuaciones necesarias es muy similar al anterior, necesitando conocer cuál es la razón de recirculación (R). De igual manera, se ha incluido la figura 2 para facilitar la comprensión de la notación utilizada.

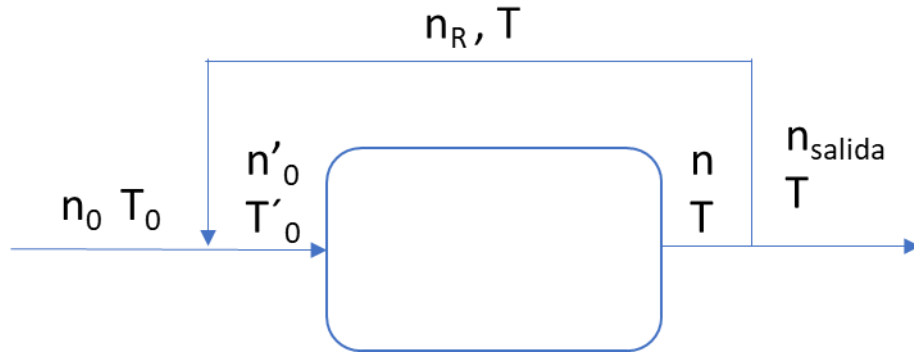


Figura 2: Esquema genérico de un reactor con una corriente de recirculación. Fuente: elaboración propia.

$$R = \frac{n_R}{n_{salida}}$$

$$n = n_{salida} + n_R = n_{salida}(1 + R)$$

$$n'_0 = n_0 + n_R = n_0 + Rn_{salida} = n_0 + \frac{R}{1 + R}n \quad (21)$$

Si, como se ha realizado para el bypass, se tiene en cuenta que la corriente de recirculación tenga una temperatura diferente a la del alimento, será necesario recalcular la temperatura a la que se encuentra la corriente con la que se alimenta directamente al reactor.

$$T'_0 = \frac{\sum_j n_{0,j} C_{p,j} T_0 + \sum_j n_{R,j} C_{p,j} T}{\sum_j (n_{0,j} + n_{R,j}) C_{p,j}} \quad (22)$$

Aunque conceptualmente la recirculación y el bypass sean muy similares, pues solo suponen una división y mezcla de corrientes, la simulación de un reactor con recirculación contiene una complicación adicional que puede intuirse a partir de las ecuaciones 21 y 22: el alimento del reactor es dependiente de la corriente de producto, que es la incógnita, aquello que se desea conocer. Por tanto, es necesario emplear un procedimiento iterativo en el que se parte de unos valores supuestos de moles y temperatura a la salida del reactor para calcular la corriente de entrada y se comparan con los obtenidos tras la simulación hasta que los valores supuestos y calculados son coincidentes.

Nótese que en ningún momento se ha hecho referencia a la presión. A modo de simplificación se ha considerado que, para el caso del bypass, la corriente de producto final está a la misma presión que la corriente de salida del reactor. Cuando se trata de una recirculación, se asume que antes del mezclador existe un compresor que iguala la presión de la corriente de recirculación a la de la corriente de alimentación del sistema con el objetivo de que la pérdida de carga ocasionada por el reactor quede compensada. Por tanto, la corriente de alimentación del sistema y la de recirculación se encuentran a la misma presión al entrar al mezclador.

6.6 Asociación de reactores

La simulación de un conjunto de reactores en paralelo presenta ciertas similitudes con el tratamiento realizado en el punto anterior para un reactor que incluye una corriente de bypass. Partiendo de una corriente de alimento global para todo el sistema, esta se reparte en un divisor de corrientes para dar lugar a cada una de las corrientes de entrada a cada uno de los reactores, como se muestra en la figura 3.

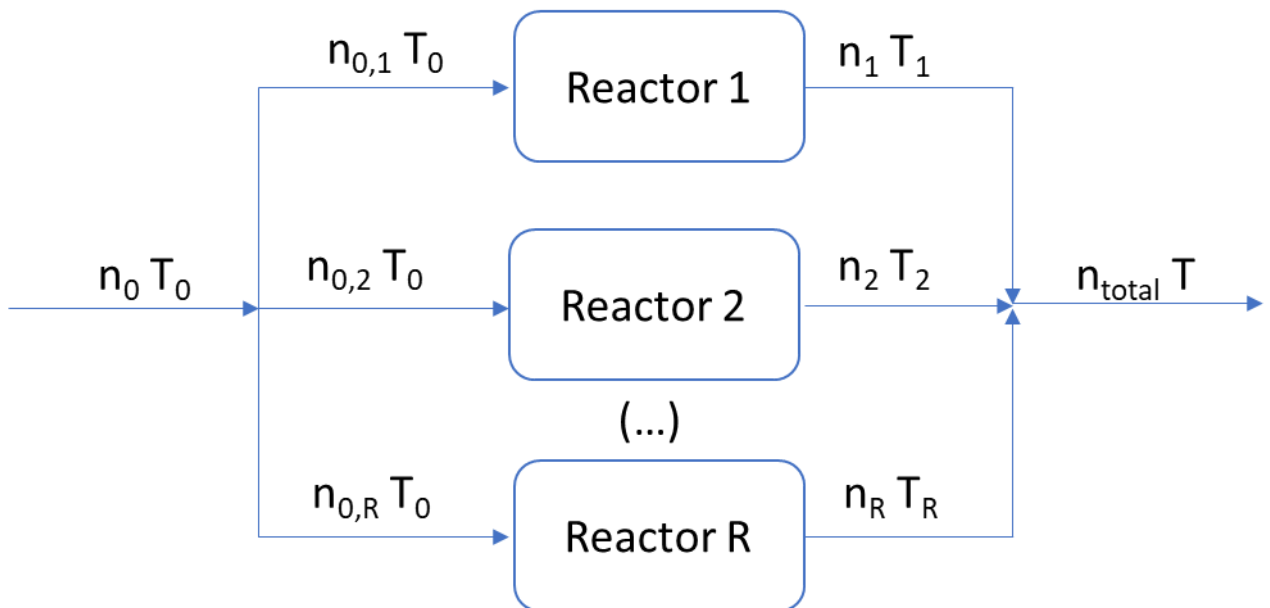


Figura 3: Esquema genérico para un conjunto de reactores en paralelo. Fuente: elaboración propia.

Para conocer el caudal molar de entrada a cada reactor, de forma predeterminada se asume que la corriente de alimento se reparte de forma equitativa, aunque también puede especificarse la fracción de la corriente de alimento que se lleva a cada reactor, designada como α en la ecuación 23. Posteriormente, se simula cada uno de los reactores de forma independiente y se realiza un balance de materia y energía en el mezclador para calcular la corriente de producto total. El valor asignado a la presión de la corriente final, es decir, la que sale del mezclador, es el de la presión más baja de todas las corrientes que se combinan.

Las ecuaciones que rigen este proceso se indican a continuación.

$$n_{0,r} = \alpha_r n_0 \quad (23)$$

$$n_{total} = \sum_r n_r \quad (24)$$

$$T = \frac{\sum_r \sum_j n_{j,r} C_{p,j} T_r}{\sum_r \sum_j n_{j,r} C_{p,j}} \quad (25)$$

$$P = \min (P_1, P_2, \dots P_r) \quad (26)$$

En las expresiones anteriores el subíndice j sigue haciendo referencia a cada uno de los compuestos presentes en el sistema y el subíndice r representa cada uno de los reactores que se combinan.

La forma de proceder para asociaciones en serie es más sencilla pues solo hay que tener en cuenta que el producto de un reactor constituirá el alimento del siguiente.

6.7 Optimización del reactor

Utilizando **ReactorApp** puede estimarse el tiempo de residencia óptimo necesario para conseguir que el producto cumpla con una serie de requisitos. Se trata por tanto de problemas con un fuerte carácter no lineal cuya resolución se ha llevado a cabo recurriendo al comando *fmincon* de MATLAB. Para abordar esta clase de cuestiones, tanto el alimento como las características del reactor deben ser conocidos. A pesar de que cualquier corriente queda totalmente definida si se conoce su composición, temperatura y presión, en la formulación del problema puede que no se especifiquen todas estas variables para la corriente de producto, sino que solo interese fijar algunas de ellas. Por ejemplo, puede especificarse que la producción del compuesto deseado alcance un determinado valor sin atender a la cantidad producida del resto de especies. En la formulación del problema, la cual se detalla a continuación para cada uno de los reactores, las variables de la corriente cuyo valor se ha indicado expresamente se identifican con un asterisco y utilizando la letra mayúscula Y, con el objetivo expresar de forma genérica dicha variable fijada. Por tanto, Y puede ser tanto la temperatura, como la presión o el caudal molar de una determinada especie.

- Reactor de flujo pistón (RFP): la variable a optimizar es la longitud del reactor por lo que se tiene en cuenta que el número de tubos y su sección son conocidos.

$$\begin{aligned}
& \min_L \quad V = S(z)L \\
& \text{s.a,} \\
& \frac{dn_j}{dz} = f_j^{BM}(n_j, T, P, Q_v), \quad \forall j \\
& \frac{dT}{dz} = f^{BE}(n_j, T, P, Q_v) \\
& \frac{dP}{dz} = f^{BCM}(n_j, T, P, Q_v) \\
& f^{Ec. estado}(n_j, T, P, Q_v) = 0 \\
& Y(z = L) = Y^*
\end{aligned} \tag{27}$$

- Reactor de tanque agitado (RCTA): para este tipo de reactores la variable que se optimiza es el volumen del reactor y la formulación es muy similar a la empleada para el RFP, con la diferencia que se asume que no existen pérdidas por fricción y, por tanto, la presión es constante en toda el sistema (la corriente de alimentación y de salida se encuentran a la misma presión).

$$\begin{aligned}
& \min_V \quad V \\
& \text{s.a,} \\
& f_j^{BM}(n_j, T, Q_v) = 0, \quad \forall j \\
& f^{BE}(n_j, T, Q_v) = 0 \\
& f^{Ec. estado}(n_j, T, Q_v) = 0 \\
& Y(V) = Y^*
\end{aligned} \tag{28}$$

- Reactor discontinuo de tanque agitado (RDTA): la variable de optimización en este caso no está relacionada con las dimensiones del reactor, las cuales se considera que toman un valor fijo y conocido, sino que lo que interesa minimizar es el tiempo que la mezcla reaccionante está en el interior del tanque, es decir, el tiempo de reacción (t_{reacc}).

$$\begin{aligned}
& \min_{t_{reacc}} \quad t_{reacc} \\
& \text{s.a,} \\
& \frac{dN_j}{dt} = f_j^{BM}(N_j, T, P\{V\}), \quad \forall j \\
& \frac{dT}{dt} = f^{BE}(N_j, T, P\{V\}) \\
& f^{Ec. estado}(N_j, T, P\{V\}) = 0 \\
& Y(t = t_{reacc}) = Y^*
\end{aligned} \tag{29}$$

Nótese que los balances de conservación y ecuación de estado que modelan el RDTA, son función de 3 variables, ya que se debe seleccionar entre dos modos de funcionamiento del reactor: a presión o a volumen constante.

Manteniendo un planteamiento idéntico al expuesto en las líneas anteriores y modificando únicamente la función objetivo es posible determinar las dimensiones que debe tener el reactor (o el tiempo de reacción óptimo) para conseguir el producto deseado con un coste mínimo. El procedimiento seguido para la estimación económica es objeto del apartado 6.9.

6.8 Camino de temperatura óptima

Como se introdujo en apartados anteriores de este trabajo, cuando se estudian reacciones exotérmicas reversibles, otro de los enfoques que puede adoptarse para minimizar el tiempo de residencia se basa en controlar la temperatura. Para el tipo de reacciones mencionado, resulta útil recurrir a diagramas conversión frente a temperatura, sobre los cuales se trazan curvas de velocidad de reacción constante. Un ejemplo de este diagrama se muestra en la figura 4. La temperatura óptima es aquella que permite que la reacción transcurra a la máxima velocidad posible.

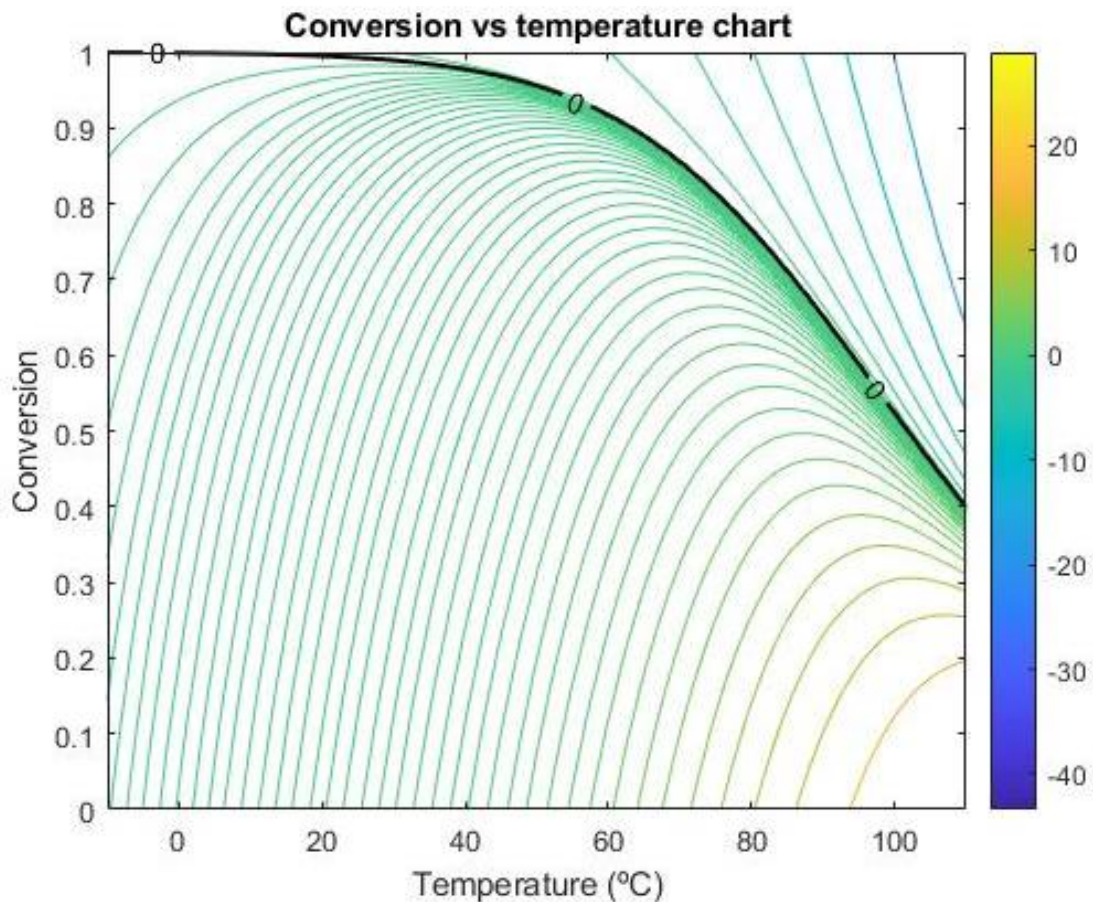


Figura 4: Diagrama conversión frente a temperatura para una reacción exotérmica reversible obtenido con [ReactorApp](#). Fuente: elaboración propia.

Si el reactor estudiado es un RCTA, como el efluente y la mezcla dentro del reactor tienen la misma composición, la temperatura óptima de operación será aquella que proporcione la máxima velocidad de reacción para la conversión final deseada. Por consiguiente, el mínimo tiempo de residencia vendrá dado por la ecuación 30.

$$\tau = \frac{V}{Q_{v0}} = \frac{1}{Q_{v0}} \cdot \frac{n_{k0} X_k}{-r_{j,m\acute{a}x}} \quad (30)$$

Para hallar el mínimo tiempo de residencia en un RFP o un RDTA utilizando este tipo de representaciones los pasos que se deben seguir son:

1. Trazar sobre un diagrama conversión frente a temperatura el perfil de temperatura óptimo que debe seguir el reactor.
2. Registrar los valores de conversión y velocidad de reacción a lo largo de dicho camino.
3. Obtener el tiempo de residencia mínimo a partir de una integración numérica de la inversa de la velocidad de reacción respecto a la conversión del reactivo limitante. Las ecuaciones que se emplean para cada tipo de reactor se muestran a continuación.

- RDTA:
$$t = C_{k0} \int_0^{X_k} \frac{dX_k}{-r_k} \quad (31)$$

- RFP:
$$\tau = \frac{V}{Q_{v0}} = \frac{n_{k0}}{Q_{v0}} \int_0^{X_k} \frac{dX_k}{-r_k} \quad (32)$$

En la secuencia anterior, el primer paso será diferente en función de cómo se regule la temperatura del reactor. En un reactor no isoterma, la situación ideal es aquella en la que la temperatura evoluciona siguiendo el camino que une los máximos de las curvas, como se trata de ilustrar con la línea roja de la figura 5, ya que en todo momento opera a la máxima velocidad de reacción.

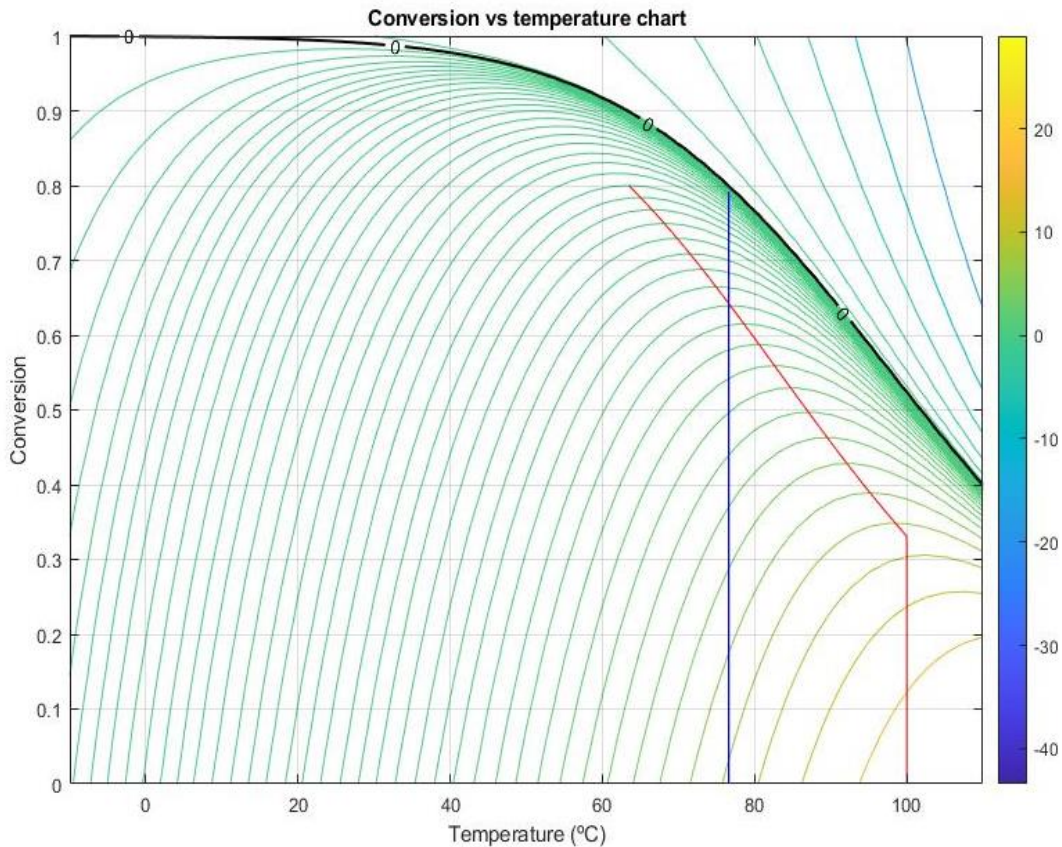


Figura 5: Evolución de la temperatura siguiendo el camino de máxima velocidad de reacción dibujada con [ReactorApp](#). Fuente: elaboración propia.

Hallar este camino supone resolver un problema de optimización para cada valor de conversión comprendido entre 0 y la conversión final deseada. En cada uno de dichos puntos, se fija el valor de la conversión de tal manera que la velocidad de reacción pasa a depender únicamente de la temperatura. El objetivo del problema de optimización es encontrar la temperatura a la cual la velocidad de reacción es máxima. Una formulación matemática de dicho enunciado es la siguiente:

$$\begin{aligned} \max_T \quad & r_i(T) \\ \text{s. a.}, \quad & T_{LI} \leq T \leq T_{LS} \end{aligned} \quad (33)$$

Es importante señalar que se ha acotado la temperatura a un intervalo comprendido entre un límite inferior (T_{LI}) y uno superior (T_{LS}) para facilitar la convergencia y que se ha utilizado el subíndice i para denotar que se trata de la velocidad referida a la reacción. Si se hubiera utilizado la velocidad de reacción referida a un compuesto (r_j) habría sido necesario cambiar el sentido de la optimización dependiendo de si se tratara de un producto (r_j tomaría valores positivos y habría que maximizar) o un reactivo (r_j tomaría valores negativos porque se estaría consumiendo y debería minimizarse).

Cuando se trata de reactores isotermos, la trayectoria seguida por el reactor queda representada por la línea azul de la figura 5. La temperatura de operación óptima es aquella que permita trabajar a la máxima velocidad de reacción media durante el transcurso de la reacción (hasta un valor final de la conversión deseado X_k^{final} , que puede calcularse aplicando el teorema del valor medio del cálculo integral.

$$\bar{r}_{i,máx}(T) = \frac{1}{X_k^{final}} \int_0^{X_k^{final}} r_i(X_k, T) dX_k \quad (34)$$

La formulación del problema de optimización para averiguar dicha temperatura incluye una nueva restricción para asegurar que la trayectoria seguida no cruza el equilibrio, lo cual sería físicamente imposible.

$$\begin{aligned} \max_T \quad & \bar{r}_{i,máx}(T) = \frac{1}{X_k^{final}} \int_0^{X_k^{final}} r_i(X_k, T) dX_k \\ \text{s.a,} \quad & \\ & r_i(X_k^{final}, T) \geq 0 \\ & T_{LI} \leq T \leq T_{LS} \end{aligned} \quad (35)$$

6.9 Estimación económica

6.9.1 Coste del capital (CAPEX)

Para realizar una determinar el coste del capital se va a seguir el método descrito en el Anexo I del libro de Richard Turton (2018). Dicho método, originalmente introducido por Guthrie, se considera adecuado para estimar el coste de los equipos en una etapa preliminar del diseño de una planta, es decir, para ofrecer una aproximación de la inversión económica que se debe realizar, lo cual resulta especialmente útil para comparar entre diferentes alternativas.

En esta técnica, se calcula el coste de compra en unas condiciones de partida y, por medio de una serie de factores multiplicadores que dependen de la presión de operación, del material o del tipo de equipo se evalúa el coste final de los bienes.

Para determinar el coste de compra del equipo operando a presión moderada y utilizando acero al carbono como material de construcción (C_P^0) se utiliza la siguiente expresión.

$$\log C_P^0 = K_1 + K_2 \log(Q) + K_3 (\log(Q))^2 \quad (36)$$

siendo K_1 , K_2 y K_3 una serie de constantes que dependen del tipo de equipo considerado y Q la capacidad del reactor en m^3 o la superficie del intercambiador de calor en m^2 . Los valores que se han utilizado para estos parámetros se muestran en la tabla 1 de esta memoria y han sido extraídos de la tabla A1 del mencionado libro de Turton, considerando los RFP como *Horizontal Process Vessels*, los RCTA o RDTA como *Jacketed Agitated Reactors* y los intercambiadores de calor del tipo *Fixed tubes*. La ecuación 36 proporciona el coste expresado en dólares en base al año 2001.

A partir del coste de compra, se puede evaluar el coste total del equipo (C_{BM}), que representa la suma de los costes directos, como los materiales y las labores de instalación, y de los indirectos, como el seguro, el transporte o los beneficios del contratista.

$$C_{BM} = C_P^o F_{BM} \quad (37)$$

Para reactores de tanque agitado, el factor multiplicador (F_{BM}) toma el valor de 4, tal y como se indica en la tabla A7 del Apéndice I del mencionado libro. En el caso de reactores de flujo pistón, el factor F_{BM} se calcula por medio de la ecuación 38.

$$F_{BM} = B_1 + B_2 F_M F_P \quad (38)$$

En la expresión anterior, B_1 y B_2 son dos constantes que dependen del tipo de equipo considerado y cuyo valor se indica en la tabla 1. Se considera que en todos los casos el material para la construcción de los equipos es acero al carbono, por lo que el factor material (F_M) es igual a la unidad. Finalmente, el factor de presión (F_P) para intercambiadores de calor se asume como 1 y, para reactores de flujo pistón, puede calcularse en función de la presión manométrica de operación (P) expresada en barg y del diámetro del reactor (D) en metros.

$$F_P = \frac{\frac{PD}{2[850 - 0,6 P]} + 0,00315}{0,0063} \quad (39)$$

Si el valor obtenido es menor de 1, F_P toma el valor de 1. En cambio, si la presión de operación es inferior a -0,5 barg, el factor de presión se considerará igual a 1,25.

Tabla 1: Constantes para la estimación del coste de los reactores e intercambiadores de calor.

	RFP	RCTA y RDTA	Intercambiadores
K1	3,5565	4,1052	4,3247
K2	0,3776	0,5320	-0,3030
K3	0,0905	-0,0005	0,1634
B1	1,49	-	1,63
B2	1,52	-	1,66

Nota. Adaptado de “*Analysis, Synthesis and Design of Chemical Processes*”, de Turton, R., 2018, Prentice Hall.

El resultado obtenido por este procedimiento proporciona el coste referido al año 2001, por lo que es necesario actualizarlo. Para ello se han utilizado los índices CEPCI (2020) correspondientes a los años 2001 y 2020, cuyo valor es de 397 y 595.4, respectivamente.

$$CAPEX = C_{BM} \frac{CEPCI_{2019}}{CEPCI_{2001}} \quad (40)$$

Para reactores que cuenten con un sistema de calefacción o refrigeración, el coste total del equipo será la suma entre el coste del reactor estudiado y el del intercambiador de calor correspondiente.

6.9.2 Coste operacional (OPEX)

Los servicios de calefacción y refrigeración son uno de los factores con más peso en el importe total de los costes de operación y, por este motivo, la estimación de los costes operacionales se va a realizar en base al consumo de fluido calefactor o refrigerante.

$$OPEX \left(\frac{\$}{\text{año}} \right) = Q \text{ (kW)} \cdot \text{Precio fluido} \left(\frac{\$}{\text{kW} \cdot \text{año}} \right) \quad (41)$$

En la ecuación anterior, Q hace referencia al flujo de calor intercambiado entre el reactor y el sistema de control de temperatura.

6.9.3 Coste total anualizado (TAC)

El importe total del equipo se obtiene a partir del CAPEX y el OPEX. Ambas cantidades deben expresarse en las mismas unidades para poder sumarlas y por eso, el CAPEX debe multiplicarse por un factor de anualización (F) que tiene unidades de años⁻¹.

$$TAC = OPEX + CAPEX \cdot F$$

El factor de actualización se puede calcular estableciendo un tiempo de vida útil del equipo (n) y un tipo de interés (i). Se han fijado estos valores en 10 años y 0,1, respectivamente.

$$F = \frac{i(1+i)^n}{(1+i)^n - 1} \quad (42)$$

7 METODOLOGÍA

La *toolbox* **ReactorApp** está estructurada como un conjunto de funciones y clases de MATLAB, utilizadas por una interfaz gráfica de usuario para recibir la información y mostrar los resultados. Esta forma especial de programar se conoce como programación orientada a objetos (POO). Los objetos son elementos individuales que contienen una serie de información y que realizan unas acciones determinadas. Todos los objetos que comparten características similares pertenecen a una misma clase. Por tanto, una clase puede entenderse como una plantilla que define de forma general cómo son los objetos de un determinado tipo. Para ilustrar estas ideas resulta útil acudir a un ejemplo. En POO, podría definirse la clase “persona”. Un objeto o instancia de esta clase puede tener una serie de atributos como el nombre o la edad y realizar una serie de funciones. Sin embargo, una clase en sí no tiene utilidad, simplemente es una entidad abstracta. Para que sea útil hay que instanciarla, es decir, crear un objeto con características propias como John Smith de 40 años que desempeña funciones concretas como comprar o trabajar.

Siguiendo un razonamiento similar, se han dividido los conceptos que se manejan en la simulación de reactores en tres clases diferenciadas:

- **Reactor** (clase para el reactor): incluye todos los parámetros de diseño necesarios para conocer las dimensiones, forma o construcción del reactor a utilizar y habilita una serie de funciones que se aplican a los reactores (balances de materia, energía o cantidad de movimiento, asociación de reactores, cálculo del mínimo tiempo de residencia...). El archivo “Reactor.m” es el que incluye todos estos aspectos relativos a los reactores.
- **ReactionSys** (clase para el sistema reaccionante): en esta clase se definen los compuestos presentes en el sistema objeto de estudio, indicando sus propiedades físicas y termodinámicas. También se especifican los datos asociados a las diferentes reacciones que se producen como los parámetros cinéticos, la entalpía de reacción o la estequiometría y se incluye la función que calcula la velocidad de reacción, al ser su expresión independiente del tipo de reactor empleado. El archivo donde se incluyen todas sus propiedades y funciones es “ReactionSys.m”.
- **Stream** (clase para la corriente): recoge las variables necesarias para especificar una corriente, es decir, su composición, caudal, temperatura, presión, densidad o viscosidad. En “Stream.m” queda definida esta clase.

La elección de trabajar utilizando una estrategia de programación orientada a objetos comprende una serie de ventajas. Por un lado, permite organizar el código de forma ordenada lo cual facilita

la legibilidad y comprensión del mismo, facilitando que pueda ser modificado y reutilizado por otras personas. Por otra parte, no son únicamente cuestiones de claridad las que hacen de la programación orientada a objetos la estrategia ideal para este trabajo, sino por las ventajas que ofrece en cuanto a la transmisión de información entre diferentes archivos y funciones, evitando el uso de variables globales que puedan reescribirse o verse modificadas durante la ejecución del programa.

A continuación, se describen los aspectos más importantes de las diferentes clases con las que trabaja en **ReactorApp**.

7.1 Sistema reaccionante

La clase que define el sistema reaccionante, denominada **ReactionSys**, contempla aquellos parámetros característicos de la reacción o conjunto de reacciones estudiado, así como de los componentes que en ellas intervienen o que participan como inertes. Todos ellos son independientes del tipo de reactor que se utilice o de las condiciones de operación. En este sentido, las propiedades asociadas a un sistema reaccionante se recogen en la tabla 2. La primera columna contiene el nombre con el que cada una de las ellas se ha definido en el código de MATLAB, y está acompañada de una segunda columna en la que se indica a qué hace referencia cada uno de dichos atributos.

Tabla 2: Nombre asignado y descripción de las propiedades que identifican el sistema reaccionante

Propiedad	Descripción
componentName	Nombre genérico de los compuestos que intervienen
componentFormula	Fórmula de los compuestos presentes en el sistema
componentMw	Peso molecular en g/mol
componentCp	Capacidad calorífica expresada en J/(mol·K)
componentHeatOfFormation	Entalpía de formación de cada componente en J/mol
stoichiometricMatrix	Matriz de coeficientes estequiométricos
partialOrders	Orden parcial de reacción de cada componente considerando que se trata de reacciones elementales
k0	Factor pre-exponencial y energía de activación para el cálculo de la constante cinética según la ley de Arrhenius
Ea	
k0_denominator	Factor pre-exponencial, energía de activación y órdenes de reacción de cada componente para el cálculo de cada término del sumatorio denominador de la ley cinética de LHHW
Ea_denominator	
partialOrders_denominator	

Fuente: elaboración propia.

Tabla 2: Nombre asignado y descripción de las propiedades que identifican el sistema reaccionante (Continuación)

Propiedad	Descripción
DHref	Entalpía de reacción a la temperatura de referencia (implementada como Tref), que por defecto es de 25°C (273.15 K)
phase	Estado de agregación (líquido o gas).

Fuente: elaboración propia.

Dentro del archivo de la clase del sistema reaccionante se han implementado dos funciones: una para establecer conexión con el programa Aspen HYSYS para extraer datos y otra donde se calcula la velocidad de reacción.

7.1.1 Cálculo de la velocidad de reacción

La función que realiza el cómputo de la velocidad de reacción necesita que los argumentos de entrada sean la temperatura y la concentración, expresada en moles por unidad de volumen.

Para evaluar la velocidad de reacción, de forma predeterminada se utiliza el modelo de Langmuir–Hinshelwood–Hougen–Watson (LHHW), cuya expresión se ha presentado en el apartado 6.1 del fundamento teórico.

Sin embargo, puede que se estudien otros sistemas para los cuales este modelo no describa bien su comportamiento. Por ejemplo, si se desea simular un reactor en el que se lleva a cabo una reacción enzimática, la mejor opción será recurrir a la cinética de Michaelis-Menten. Ante estas situaciones, la aplicación que se ha creado en este trabajo contempla la posibilidad de utilizar otros modelos personalizados. Para ello, el usuario podrá especificar sus propias expresiones haciendo uso de la interfaz gráfica de la aplicación o bien elaborar un archivo o script que incluya todas las ecuaciones del modelo que mejor se ajuste a su problema concreto teniendo en cuenta una serie de consideraciones:

- Las variables de entrada de la función deben ser las mismas, es decir, concentración como primer argumento y temperatura como segundo argumento.
- El resultado que devuelva la función o argumento de salida debe ser un vector fila, con tantos elementos como reacciones, por lo que cada elemento del vector contiene el valor de la velocidad de cada una de las reacciones que tienen lugar para los valores de concentración y temperatura indicados.
- Guardar la función como un archivo “.m”.

En otras palabras, por defecto se utilizará el modelo de LHHW pero además el usuario tiene total libertad para escoger la ecuación de velocidad que mejor describa el sistema reaccionante estudiado, lo cual supone una pequeña ventaja frente a otros simuladores como Aspen HYSYS.

7.1.2 Comunicación con Aspen HYSYS

Uno de los principales problemas a la hora de realizar una simulación en MATLAB es la dificultad de disponer de todos los datos necesarios para llevar a cabo los cálculos. En muchas ocasiones, el acceso a las propiedades termodinámicas es limitado, exceptuando aquellos compuestos o reacciones más frecuentes y estudiados. Resulta especialmente complicado disponer de todos los parámetros cuando además se precisa evaluar cómo afecta la temperatura a variables como la capacidad calorífica.

Una de las utilidades de este trabajo es la capacidad de importar datos termodinámicos directamente desde el programa Aspen HYSYS, utilizándolo como base de datos. De forma general, para que esta comunicación pueda llevarse a cabo, el usuario únicamente debe especificar en un archivo de Aspen HYSYS cuáles son los componentes que van a estar presentes en el proceso a simular y qué paquete termodinámico se va a emplear. Por tanto, no se requieren conocimientos profundos de este software ya que este programa se emplea como una mera fuente de consulta. En este punto, existen dos posibilidades, planteadas en función de la experiencia previa del usuario con el simulador de procesos.

Una opción pensada para aquellas personas menos familiarizadas con el uso de Aspen HYSYS es utilizar el archivo “ComponentDataBase.xml”. Este archivo se encuentra en el mismo directorio o carpeta que [ReactorApp](#). Al abrirlo, se puede comprobar que en el entorno de simulación solo hay una corriente material, sobre la que se trabajará y aplicarán modificaciones para conseguir la información de interés, lo cual se describirá posteriormente. Los pasos que deben seguirse cuando se utiliza este archivo son:

1. Abrir el archivo “ComponentDataBase.xml”
2. Desde la pestaña *Properties*, seleccionar todos los componentes que intervengan en el proceso que se pretende estudiar
3. Desde la pestaña *Fluid Package* cambiar el paquete termodinámico.
4. Guardar el archivo

Por otra parte, para aquellos con un conocimiento previo del programa puede ser más práctico utilizar un archivo ya existente sobre el que se haya trabajado y en el que ya estén incluidos todos los compuestos y el paquete termodinámico. La única condición es que esté guardado en formato “.xml” para evitar posibles problemas con la versión de Aspen HYSYS que se utilice.

En este caso, habrá que indicar cuál es el nombre del archivo, su ubicación y la corriente a partir de la cual se obtendrán los datos. Todo este procedimiento está guiado y se realiza de forma intuitiva cuando se trabaja desde la interfaz gráfica de **ReactorApp**. Es muy recomendable no utilizar una corriente directamente conectada a otros elementos (un reactor, una columna, una recirculación...) ya que, al aplicar cambios sobre ella, se verá afectada la simulación y pueden aparecer problemas de convergencia. Por tanto, lo más conveniente es trabajar sobre una corriente material independiente, sin conectar con ningún otro elemento, que puede ser una copia de cualquiera de las otras que aparezcan en el diagrama de flujo.

La función que lleva a cabo el intercambio de información entre MATLAB y Aspen HYSYS, sigue la siguiente secuencia:

- Paso 1: encontrar el archivo ".xml"
- Paso 2: establecer la conexión con Aspen HYSYS
- Paso 3: leer las propiedades accesibles de forma directa
- Paso 4: obtener el calor específico (C_p) de cada compuesto

Se ha diferenciado entre el paso 3 y 4 ya que hay propiedades que pueden obtenerse directamente de Aspen HYSYS sin realizar ningún tipo de cálculo. Estas propiedades son:

- Nombre convencional del compuesto
- Fórmula
- Peso molecular
- Entalpía de formación

Por el contrario, en el caso de la capacidad calorífica el procedimiento no es sencillo ya que, según el paquete termodinámico utilizado, el simulador de procesos Aspen HYSYS procesa de manera distinta esta propiedad termodinámica. En el paso 4 entra en juego la corriente material de la que se ha hablado anteriormente debido a que, para determinar el calor específico de cada compuesto, lo que se hace es modificar los valores de composición y temperatura de la corriente especificada y leer el valor de la capacidad calorífica resultante tras aplicar dichas modificaciones. En consecuencia, no se obtiene un único valor de C_p sino su variación bajo las distintas condiciones probadas. Ante este resultado el usuario puede decidir si quedarse con el valor medio que toma esta propiedad en el rango de temperatura que se ha evaluado o si, por el contrario, prefiere ajustar los datos a una expresión polinómica para poder tener en cuenta en los balances de energía y cantidad de movimiento la influencia de la temperatura.

Entrando un poco más en detalle, en este paso 4, la función desarrolla las siguientes acciones para cada uno de los componentes, en cuya descripción se utiliza el término de corriente de trabajo para designar a la corriente sobre la que se aplican las modificaciones:

1. Se fija que la corriente de trabajo tiene un flujo molar de 1 kmol/h del compuesto j y 0 kmol/h de los demás. En otras palabras, se establece que circula 1 kmol/h del compuesto j puro.
2. Se modifica la temperatura de la corriente de trabajo, de modo que se barren todos los valores de temperatura dentro del rango definido para esta variable, comprendido entre 500 y 1000°C. Si es que se quiere considerar también la influencia de este parámetro, lo haría entre 1 a 5 atm.
3. Se lee capacidad calorífica en kJ/(kmol·°C) para cada combinación de valores probada, de manera que dichos datos quedan almacenados en un vector.
4. Se ajustan los datos a expresiones polinómicas de segundo orden que permiten evaluar cómo varía la capacidad calorífica con la temperatura. Para el ajuste, se utiliza la función *fit* de MATLAB, eligiendo la temperatura como variable independiente y la capacidad calorífica como la variable dependiente, de modo que se obtiene una expresión del tipo:

$$C_p(T) = a T^2 + b T + c \quad (43)$$

5. Se evalúa la capacidad calorífica promedio en el rango de temperaturas estudiado. Este cálculo resulta útil cuando no se requiere gran precisión en los cálculos del balance de energía, pero es necesario recurrir a Aspen HYSYS para conocer el valor de la capacidad calorífica.

$$\overline{C_p} = \frac{\int_{T_{min}}^{T_{max}} C_p(T) dt}{(T_{max} - T_{min})} \quad (44)$$

Siguiendo una metodología análoga a la explicada en las líneas anteriores, podría evaluarse conjuntamente la influencia de la temperatura y la presión sobre la capacidad calorífica, aunque esto solo resulta necesario en casos muy puntuales en los que las presiones de trabajo sean realmente elevadas y, por este motivo, no se ha abordado con gran detalle.

Finalmente, para facilitar la comprensión de la secuencia descrita, en la figura 6 se incluye el pseudocódigo del algoritmo.

```

nComponents = número de compuestos presentes en el sistema
rangoTemperatura = vector [1 x t]
rangoPresión = vector [1 x p]
for i = 1:nComponents (i varía desde 1 hasta nComponents)
    for i2 = 1 : p
        P = elemento i2 del vector de rangoPresión
        Fijar la presión de la corriente de trabajo = P
        for i3 = 1 : t
            T = elemento i3 del vector rangoTemperatura
            Fijar la temperatura de la corriente de trabajo = T
            Cp(i3,i2) = valor de la Cp de la corriente de trabajo
            % Notar que se almacena el valor leído en el elemento
            (i3, i2) de una matriz de t filas y p columnas
        end
    end
    if p > 1 % Se quiere estudiar el efecto de la presión sobre la Cp
        Cp(T,P) = fit(rangoTemperatura,rangoPresión,Cp)
    end
    Cp(T) = fit(rangoTemperatura,Cp(:,1)) % Solo la columna de Patm
    Cp, promedio = trapz(rangoTemperatura, Cp(T)) / (Tmax-Tmin)
end
end

```

Figura 6: Pseudocódigo de la función *call_DataBase*. Fuente: elaboración propia.

En síntesis, las salidas o resultados que puede devolver esta función son:

- Propiedades leídas directamente: peso molecular, entalpía formación, etc.
- Expresión que relaciona la C_p con la temperatura o con la temperatura y la presión
- Valor medio de la C_p en el rango de temperaturas estudiado y a presión atmosférica

Por último, cabe recordar que la importación de datos desde Aspen HYSYS es una opción más al servicio del usuario y no un requisito obligatorio para el funcionamiento del programa. En muchas ocasiones, cuando se trata de problemas propuestos en clase, el enunciado suele proveer de toda la información necesaria y, por otra parte, debido al elevado precio de las licencias de este potente simulador de procesos, no es habitual que este programa esté instalado en todos los ordenadores. Por este motivo, haber desarrollado una herramienta que fuera tan dependiente de Aspen HYSYS impondría muchas limitaciones.

7.2 Reactor

Al principio de esta sección se ha definido qué es una clase en la programación orientada a objetos, pero no se ha hecho referencia al concepto de herencia. Sin embargo, resulta muy importante debido a que la herencia es un mecanismo por el cual una clase puede heredar tanto las propiedades como los métodos de otra clase. De esta manera se establece una jerarquía entre

la superclase, que es la clase cuyas características se heredan, y la subclase, que además de heredar los atributos de la clase principal puede tener propiedades particulares.

Esta breve introducción a la idea de herencia entre clases permite comprender mejor cómo se ha estructurado la aplicación en este proyecto. En primer lugar, se ha definido la clase *Reactor*, en la cual se especifican las características o propiedades aplicables a cualquier tipo de reactor, relacionadas con el tamaño del reactor, con el sistema de calefacción o refrigeración o con las propiedades del catalizador. De igual manera que como se realizó para la clase del sistema reaccionante, se incluye la tabla 3 en la que cada una de estas propiedades aparece con la denominación empleada en el código y una breve descripción.

Tabla 3: Nombre asignado y descripción de las propiedades que identifican un reactor

Propiedad	Descripción
V	Volumen del reactor
bypassRatio	Relación entre el flujo que circula por la corriente de bypass y el caudal de entrada al reactor
densityCatalyst	Densidad del catalizador
porosityCatalyst	Porosidad o fracción volumétrica de huecos del catalizador
heatMode	Modo de intercambio de calor entre el reactor y el entorno. Puede ser isotermo, adiabático o calefactado/refrigerado, lo cual se ha denominado como “Other” en la aplicación.
U	Coefficiente global de transmisión de calor
heatTransferArea	Superficie de transmisión de calor
inletUtilityTemperature	Temperatura de entrada del fluido refrigerante o calefactante
outletUtilityTemperature	Temperatura de salida del fluido refrigerante o calefactante
pressureMode	Esta propiedad permite indicar si es preciso considerar o no el balance de cantidad de movimiento.

Fuente: elaboración propia.

A la vista de la tabla 3, puede que la definición de una temperatura de entrada y de salida del fluido refrigerante resulte incoherente con las ecuaciones 8 y 14. Esta definición se ha realizado con el objetivo de abrir una vía de mejora de la aplicación que permita tener en cuenta la pequeña variación en la temperatura de los servicios fríos o calientes. Por el momento, el tratamiento realizado es simple: si el usuario solo introduce una temperatura para dicho fluido, esta será la que se utilice en las ecuaciones 8 o 14 pero, si se especifica una temperatura de entrada y otra de salida, el valor de T_w en dichas ecuaciones será el valor medio entre la entrada y la salida.

Las funciones que se han implementado en el archivo que define esta superclase, es decir, la clase *Reactor*, permiten estudiar el efecto de la asociación de varios reactores, el uso de

corrientes auxiliares y optimizar su diseño. Por tanto, contienen todas las ecuaciones que se han presentado en las secciones correspondientes del fundamento teórico, tal y como se indica en la tabla 4.

Tabla 4: Ecuaciones implementadas en cada método o función de la clase Reactor

Función	Ecuaciones implementadas
bypass	18 - 20
compute_recycling	21, 22
compute_parallel	23 – 26
minimize_cost , minmize_residenceTime	27 - 29
find_optimalTemperaturePath	30 - 35

Fuente: elaboración propia

Sin embargo, cada uno de los tipos de reactor que se ha considerado al desarrollar esta aplicación tiene características propias y expresiones diferentes para los balances. Por ejemplo, no tiene sentido especificar la longitud de un RCTA pero sí que es un parámetro de diseño importante para dimensionar un RFP. En consecuencia, se han definido tres subclases:

- **PFR** (subclase para un RPF): contempla otros atributos como el número de tubos que constituyen el reactor, su longitud o su diámetro.
- **Batch** (subclase para un RDTA): además del volumen del reactor permite especificar el tiempo transcurrido entre las operaciones de carga y descarga del reactor
- **CSTR** (subclase para un RCTA)

Por tanto, cada una de las subclases es un archivo de función o script independiente que añade esta serie de propiedades y que tiene implementados los balances de materia, energía y cantidad de movimiento adecuados a cada configuración de reactor. Por motivos de claridad, se ha incluido la tabla 5 en la que se aclara qué expresiones de las que se introdujeron en el fundamento teórico de este trabajo están implementadas en cada una de las subclases.

Tabla 5: Ecuaciones implementadas en cada subclase

Subclase	Ecuaciones implementadas
Batch	14 - 17
CSTR	12 -14
PFR	6 - 11

Fuente: elaboración propia

7.3 Corriente

El archivo de clase que se utiliza para definir una corriente contempla las propiedades que se indican en la tabla 6.

Tabla 6: Nombre asignado y descripción de las propiedades que definen una corriente.

Propiedad	Descripción
phase	Fase o estado de agregación en el que se encuentran las sustancias presentes en la corriente. Se contemplan dos opciones: líquido o gas.
molarFlow	Flujo molar de cada componente
concentration	Concentración de cada componente
volumetricFlow	Caudal volumétrico
P	Presión
T	Temperatura
density	Densidad
viscosity	Viscosidad
streamMolarEnthalpy	Entalpía
streamHeatCapacity	Calor específico molar

Fuente: elaboración propia.

Los grados de libertad asociados a una corriente material son igual a la suma del número de componentes más 2, tal y como se deriva del teorema de Duhem. Por tanto, no es necesario especificar todas las propiedades que se han listado anteriormente para poder proceder.

Se ha incluido la opción de leer las propiedades de una corriente directamente desde Aspen HYSYS con el objetivo de agilizar el proceso de introducir todos los datos. La comunicación con este programa se lleva a cabo tal y como se ha detallado en el apartado 7.1.2 de esta memoria.

7.4 Diseño de la interfaz gráfica

La intención de completar el trabajo con una interfaz gráfica no es meramente estética, sino que lo que realmente se pretende es que el usuario pueda interactuar de forma mucho más intuitiva con la aplicación tanto a la hora de proporcionar los datos de partida como de interpretar el resultado. De esta manera, es posible hacer un uso adecuado sin necesidad de profundizar en cómo funciona el código internamente, evitando así los posibles errores derivados de una introducción incorrecta de los datos o las dificultades a la hora de visualizar la solución.

Para construir esta interfaz, conocida como *toolbox* en inglés, se ha utilizado App Designer, que es el entorno de MATLAB para crear aplicaciones. Partiendo de una plantilla en blanco como la

que se muestra en la figura 7, en la parte de la izquierda de la ventana aparecen todos los módulos u objetos que se pueden utilizar: botones, desplegables, paneles...

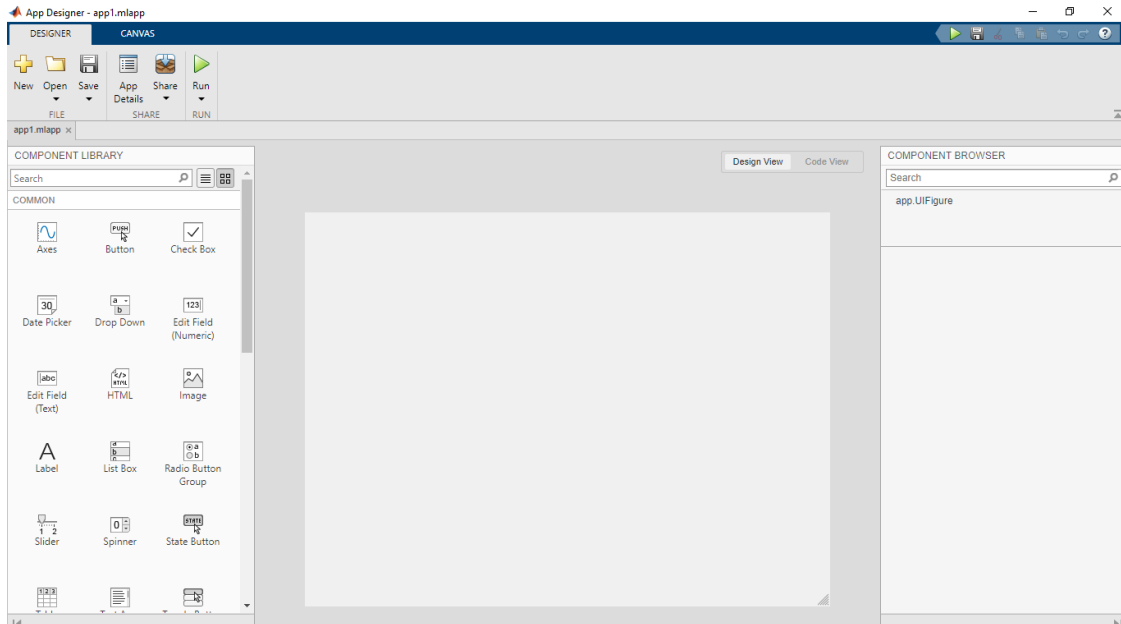



Figura 7: Plantilla en blanco para diseñar una aplicación en el entorno de App Designer. Fuente: elaboración propia.

Para conseguir el diseño deseado, basta con pinchar y arrastrar dichos elementos sobre la plantilla. Una vez hecho esto, a cada uno de los objetos seleccionados se le puede asignar una función de retrollamada o devolución de llamada (del inglés *Callback*), es decir, un conjunto de instrucciones que determinan qué ocurre cuando ese objeto es activado desde la interfaz. En la figura 8 se muestra un ejemplo muy sencillo de una devolución de llamada que se ha extraído del código de este proyecto. En este caso, la función de retrollamada “UtilitiesButtonPushed” establece que, cuando se pulsa sobre el icono , se abra la ventana “UtilitiesApp” que contiene información sobre diferentes fluidos que se pueden utilizar para calefactar o refrigerar el reactor y cuya apariencia se muestra en la sección de resultados de este trabajo, concretamente en la figura 16.

```
% Button pushed function: UtilitiesButton
function UtilitiesButtonPushed(app, event)
    app.DialogApp = UtilitiesApp(app) ;
end
```

Figura 8: Ejemplo de una devolución de llamada en App Designer. Fuente: elaboración propia.

8 RESULTADOS

El resultado de este trabajo es una aplicación multiventana que, utilizando un conjunto de funciones externas que ejecutan todos los cálculos, balances e instrucciones necesarias, permite resolver problemas de reactores. En este apartado se comentan las nociones básicas y la secuencia de pasos recomendada para utilizar la aplicación **ReactorApp**.

Antes de comenzar a trabajar es recomendable cerciorarse de que la carpeta de trabajo elegida es la adecuada, aquella en la que se encuentran todos los archivos necesarios que componen la aplicación. Una vez realizadas todas las comprobaciones pertinentes, existen dos opciones para abrir la aplicación: escribir “**ReactorApp**” en la ventana de comandos de MATLAB o seguir la secuencia Archivo > Abrir > **ReactorApp**.

Desde pantalla principal de la aplicación (figura 9) se pueden ver los diferentes problemas que es posible resolver, y la parte inferior de la ventana se va adaptando al tipo de problema seleccionado, apareciendo diversas casillas que indican qué datos es necesario introducir.

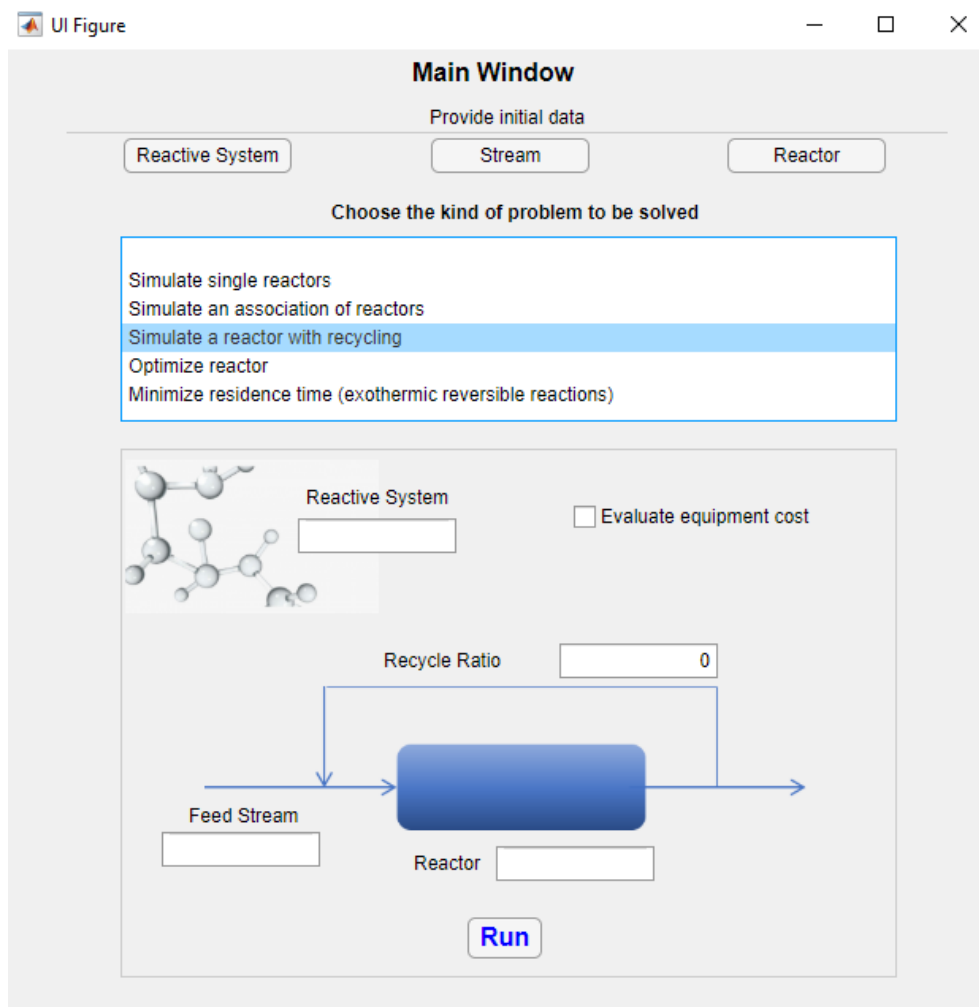


Figura 9: Apariencia de la ventana principal de **ReactorApp**. Fuente: elaboración propia.

El siguiente paso es, por tanto, introducir los datos del problema. En este aspecto, se han diferenciado tres bloques: datos relativos al sistema reaccionante (*Reactive System*), caracterización del reactor o reactores a emplear (*Reactor*) y definición de las corrientes de entrada o salida del sistema (*Stream*). Al hacer clic sobre los botones correspondientes, se abre una nueva ventana con una serie de campos que pueden completarse con la información conocida.

Tras pulsar sobre el botón *Reactive System*, la ventana que aparece contiene tres pestañas: *Stoichiometry*, *Kinetics* y *Thermodynamics*. Desde la primera de ellas (*Stoichiometry*) se introduce la matriz de coeficientes estequiométricos. Para ello, es necesario especificar previamente el número de reacciones y de compuestos que intervienen, tal y como se observa en la figura 10.

The screenshot shows the 'Reactive System' window with the 'Stoichiometry' tab selected. At the top, there is a 'Unit conversion helper' button and the title 'Reactive System'. Below the title are three tabs: 'Stoichiometry', 'Kinetics', and 'Thermodynamics'. A 'Name' input field is followed by the instruction 'Choose a name to store the data in the workspace'. The main section is titled 'Stoichiometric coefficient matrix' with the subtitle '[reactions x components]'. Below this, there are two spinners: 'Number of reactions' set to 2 and 'Number of components' set to 3. A table is provided for entering the matrix coefficients:

	A	B	C
Reaction 1			
Reaction 2			

At the bottom of the window is a 'Create Reactive System' button.

Figura 10: Aspecto de la pestaña *Stoichiometry* para introducir la matriz de coeficientes estequiométricos.
Fuente: elaboración propia.

En cuanto a la cinética de las reacciones, cuando se selecciona la pestaña *Kinetics*, la opción seleccionada por defecto es la que utiliza el modelo de LHHW. Como puede verse en la figura 11, no es necesario proporcionar el orden parcial de reacción de cada uno de los componentes puesto que se asume que las reacciones son elementales y este cálculo se realiza internamente a partir de la matriz de coeficientes estequiométricos. En el caso de querer añadir más términos al denominador de la velocidad de reacción, clicando sobre el botón *Add* se habilitará un nuevo espacio para indicar el valor de cada una de las constantes y parámetros necesarios.

Unit conversion helper **Reactive System**

Stoichiometry Kinetics Thermodynamics

Langmuir-Hinshelwood Rate Law
 Other kinetics

$$r = \frac{Num}{Den}$$


$$Num = k_0 \cdot \exp\left(-\frac{E_a}{RT}\right) \cdot \prod C_i^n$$

$$Den = 1 + \sum k'_0 \cdot \exp\left(-\frac{E'_a}{RT}\right) \cdot \prod C_i^{n'}$$

Arrhenius Rate Law constants
 Numerator of the rate expression

	k0	Ea
Reaction 1		
Reaction 2		

Constants to define the denominator

Add  Push to add a new term in the sum of the denominator

k0_denominator	Ea_denominator	P.O.Comp 1	P.O.Comp 2	P.O.Comp 3

Figura 11: Aspecto predeterminado de la pestaña *Kinetics* para utilizar el modelo de LHHW.
 Fuente: elaboración propia.

Cuando se prefiere especificar otro modelo para calcular la velocidad de reacción, debe elegirse la opción *Other kinetics*. En este caso, en la parte baja de la pantalla aparece una tabla donde pueden introducirse manualmente las ecuaciones de velocidad o bien indicar el nombre del archivo de función donde se hayan definido estos cálculos. Tomando cualquiera de estas dos alternativas, deben tenerse en cuenta ciertas pautas para escribir las expresiones correctamente:

- La temperatura debe expresarse con la letra T.
- La concentración del compuesto *j* debe escribirse como “concentration (*j*)” donde, en este contexto, *j* simboliza un número entero que puede tomar cualquier valor entre 1 y el número total de componentes. Cada valor de *j* está asociado a un compuesto en particular, atendiendo a la columna que ocupe en la matriz de coeficientes estequiométricos.
- Cuando se utiliza la tabla, pueden emplearse los operadores básicos (+, -, /, ^, * ...) pero no hay que escribir “=” al inicio de cada línea de la tabla.

La sintaxis empleada en la figura 12 sería un ejemplo adecuado de estas pautas.

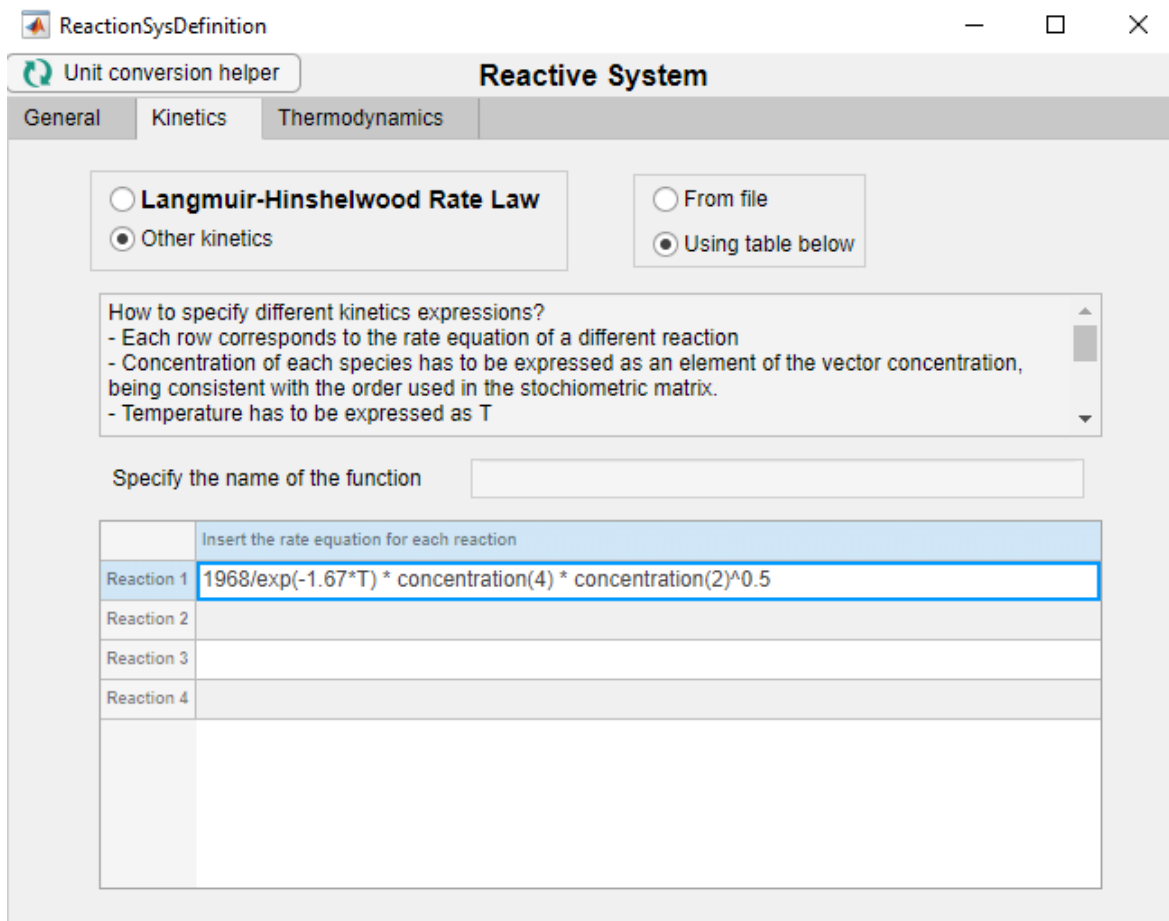


Figura 12: Aspecto de la pestaña *Kinetics* cuando se elige introducir manualmente la cinética de reacción.
 Fuente: elaboración propia.

Para suministrar los datos sobre las propiedades termodinámicas, es necesario situarse en la pestaña *Thermodynamics*, donde inicialmente solo aparece el desplegable *Property Selection* en el que se contemplan dos opciones:

- Si se tienen datos sobre el calor específico y peso molecular de los compuestos y de la entalpía de cada una de las reacciones, el usuario puede introducir directamente estos valores eligiendo la opción *User defined*.
- En casos donde esta información sea necesaria pero no se disponga de ella, puede seleccionarse la opción *Import from HYSYS*, que permite utilizar el programa Aspen HYSYS como una base de datos.

Cuando se selecciona *User defined*, en la parte inferior aparecen dos tablas, tal y como se aprecia en la figura 13, para introducir las propiedades mencionadas en la primera opción (apartado “a”) siguiendo el mismo orden de componentes que se ha utilizado en la matriz de coeficientes estequiométricos.

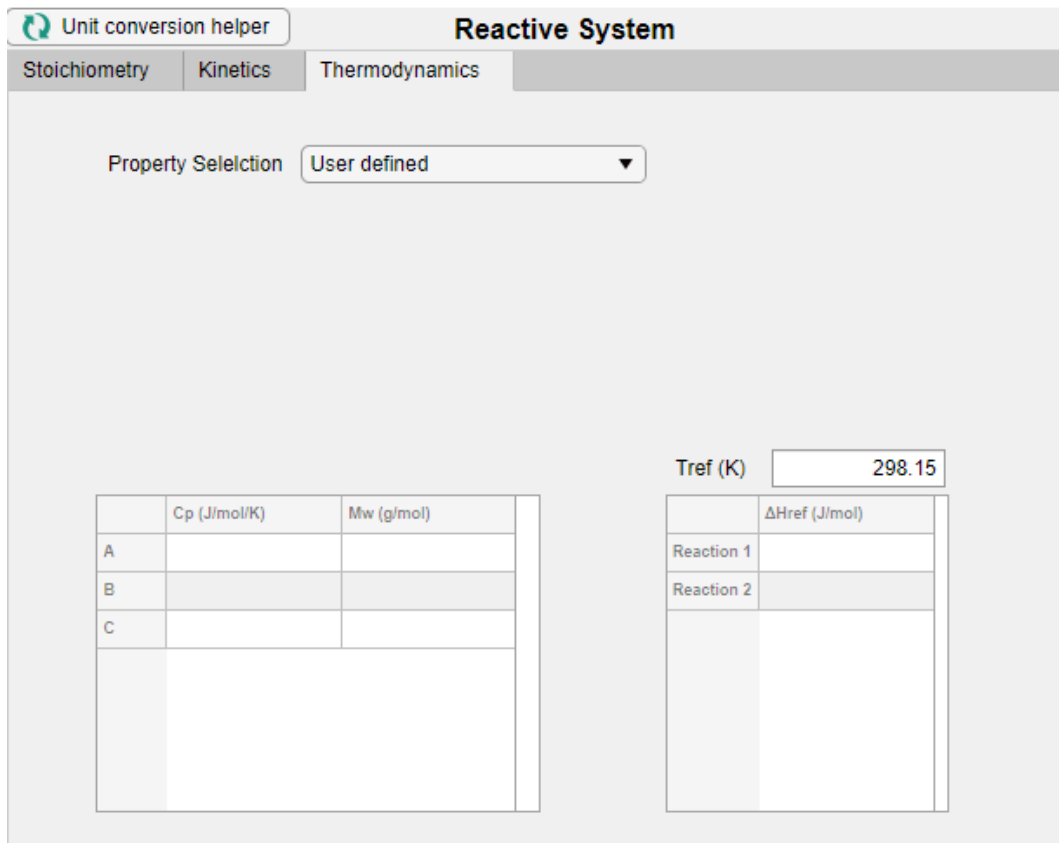


Figura 13: Aspecto de la pestaña *Thermodynamics* cuando se elige introducir manualmente las propiedades termodinámicas o el peso molecular. Fuente: elaboración propia.

En cambio, cuando se prefiere establecer una comunicación con HYSYS, es necesario indicar cuál es el archivo del que se van a obtener los datos (figura 14) y elegir desde el desplegable C_p *Option* cómo procesar la información obtenida desde HYSYS, es decir, asumir que la capacidad calorífica se mantiene constante en el intervalo de temperaturas estudiado (opción *Average*), tener en cuenta su variación con la temperatura (opción $C_p=f(T)$) o con la temperatura y la presión (opción $C_p=f(T,P)$).

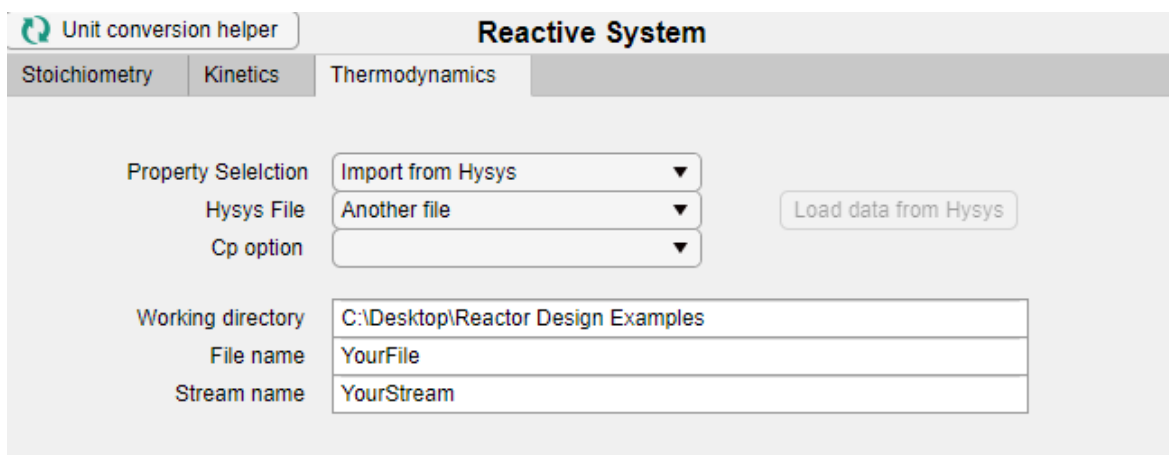


Figura 14: Ejemplo de cómo indicar el archivo para conectar con Aspen HYSYS. Fuente: elaboración propia.

Cabe destacar que no es necesario proporcionar todos los parámetros mencionados en líneas anteriores. Si el ejercicio que se pretende resolver, por ejemplo, no especificase la energía de activación en el enunciado, se podrá continuar sin ningún inconveniente.

Antes de acabar y presionar el botón *Create Reactive System* es fundamental haber indicado un nombre en el apartado *Name*, ya que es necesario para crear la instancia de la clase *ReactionSys* en la que se almacenan toda la información.

A la hora de introducir los datos referentes al reactor y a las corrientes de alimento o producto del sistema, se ha de seguir un procedimiento muy similar al que se ha descrito para las propiedades del sistema reaccionante por lo que, para evitar que este texto resulte repetitivo, solo se comentarán los aspectos más diferentes de estos bloques.

Para definir cómo es el reactor con el que se desea realizar la simulación, al pinchar sobre el botón *Reactor* de la ventana principal de la aplicación, se abre un nuevo cuadro de diálogo como el que aparece en la figura 15.

The screenshot shows a dialog box titled "Reactor" with the following elements:

- Name:** A text input field with the placeholder "Choose a name to identify the reactor".
- Display plots:** A toggle switch currently set to "off".
- Type of reactor:** Three radio buttons: "PFR", "CSTR", and "Batch".
- Volume:** A text input field with the value "0".
- Units:** A text input field.
- Bypass Ratio:** A text input field with the value "0".
- Heat exchange mode:** Three radio buttons: "Isothermal", "Adiabatic", and "Other" (which is selected).
- Does pressure change inside the reactor?:** Two radio buttons: "Constant" (selected) and "Non constant".
- How to compute pressure drop?:** Two radio buttons: "Pipe" (selected) and "Ergun".
- Utilities Data Base:** A section with the formula $Q = UA (T - Tw)$ and three input fields: "U (W/m^2/K)" with value "0", "A (m^2)" with value "1", and "Tw,inlet (K)" with value "0.1".
- Mark if the reactor is catalytic:** A checkbox that is currently unchecked.
- Density (kg/m^3):** A text input field with the value "0".
- Porosity:** A text input field with the value "0".
- Create Reactor:** A large button at the bottom center.

Figura 15: Aspecto inicial del cuadro de diálogo para introducir los datos sobre el reactor. Fuente: elaboración propia.

En el apartado donde se indica cómo es el intercambio de calor entre el reactor y el entorno, cuando este no sea ni adiabático ni isoterma (opción *Other*), puede recurrirse a una pequeña base

de datos con diferentes servicios calientes y fríos ya definidos, como se muestra en la figura 16, a la cual se accede desde el icono con una nube.

Hot Utilities

	Tin(°C)	Tout(°C)	U (W/m²/K)	Cost (\$/kW/year)	Select
Low pressure steam	125	124	800	65.1304	<input type="checkbox"/>
Medium pressure steam	175	174	800	75.4130	<input type="checkbox"/>
High pressure steam	250	249	800	85.6957	<input type="checkbox"/>
Hot oil	280	250	250	120.0000	<input type="checkbox"/>

Cold Utilities

	Tin(°C)	Tout(°C)	U (W/m²/K)	Cost (\$/kW/year)	Select
Cooling water	20	25	1000	7.2826	<input type="checkbox"/>
Air	30	35	100	0	<input type="checkbox"/>
Refrigerant fluid	-25	-24	900	93.8804	<input type="checkbox"/>

OK

Figura 16: Ventana de apoyo para la selección de los servicios fríos o calientes. Fuente: elaboración propia.

Por otra parte, cuando se acciona el botón con la palabra en inglés *Stream* desde la ventana principal, se abre una pantalla como la que aparece en la figura 17, donde se pueden introducir las variables necesarias para definir una corriente, bien sea de entrada o alimentación o de salida o producto.

UI Figure
— □ ×

Unit conversion helper
Stream
Define stream from Hysys

GUIDE: If a PRODUCT stream is being defined and you don't want to specify :

- P or T: select the check box.
- The composition of any species: simply don't enter any data in the corresponding cell

Number of components

Molar Flow	Concentration	

Molar Flow units

Name

Phase

P (Pa)

T (K)

Volumetric Flow

Units

Density

Viscosity

Create Stream

Figura 17: Apariencia de la ventana para definir una corriente de alimento o producto. Fuente: elaboración propia.

Cuando la corriente se trate de una alimentación no es necesario tener consideraciones particulares, el usuario simplemente debe rellenar todos los campos de los que disponga información o hacer clic sobre la opción *Define Stream From HYSYS* y pulsar *Create Stream* para crear una corriente con el nombre que se ha indicado en el cuadro de *Name*.

No obstante, cuando se desea optimizar el tamaño o tiempo de residencia del reactor, puede que el producto deba cumplir con unos determinados requisitos, que formarán parte de las restricciones del problema de optimización no lineal. Ante esta situación, la forma de proceder es creando una corriente de producto, igual que se ha hecho con las de alimentación, especificando únicamente los datos conocidos. Por tanto, hay que tener una serie de consideraciones:

- Si el planteamiento del problema solo incluye restricciones sobre algunos componentes solo deben rellenarse las casillas asociadas a dichos compuestos y el resto, es decir, aquellos compuestos sobre los cuales no se especifica nada en concreto, deben permanecer en blanco.
- Cuando se quiera fijar la temperatura o la presión de la corriente de salida, se deberán marcar las casillas a la izquierda de estos campos con el fin de evitar que se le asignen valores por defecto (1 atm y 25°C).

En cuanto a las unidades que se deben utilizar, no hay ningún tipo de restricción sobre el formato en el que se deben introducir, siendo el propio usuario quien debe encargarse de ser coherente con el sistema de unidades empleado. En este sentido, es importante tener en cuenta que, para reacciones en fase gas, la constante de los gases se utiliza en unidades del SI ($R = 8,314 \text{ Pa}\cdot\text{m}^3/\text{mol}\cdot\text{K}$). Para tratar de compensar este inconveniente y minimizar los errores ligados a los cambios de unidades, en la parte superior izquierda de todas las pantallas mencionadas anteriormente aparece un pequeño botón con la etiqueta *Unit conversion helper*, que da acceso a un conversor de unidades. En la figura 18 puede observarse que la interfaz creada para esta herramienta es muy sencilla.

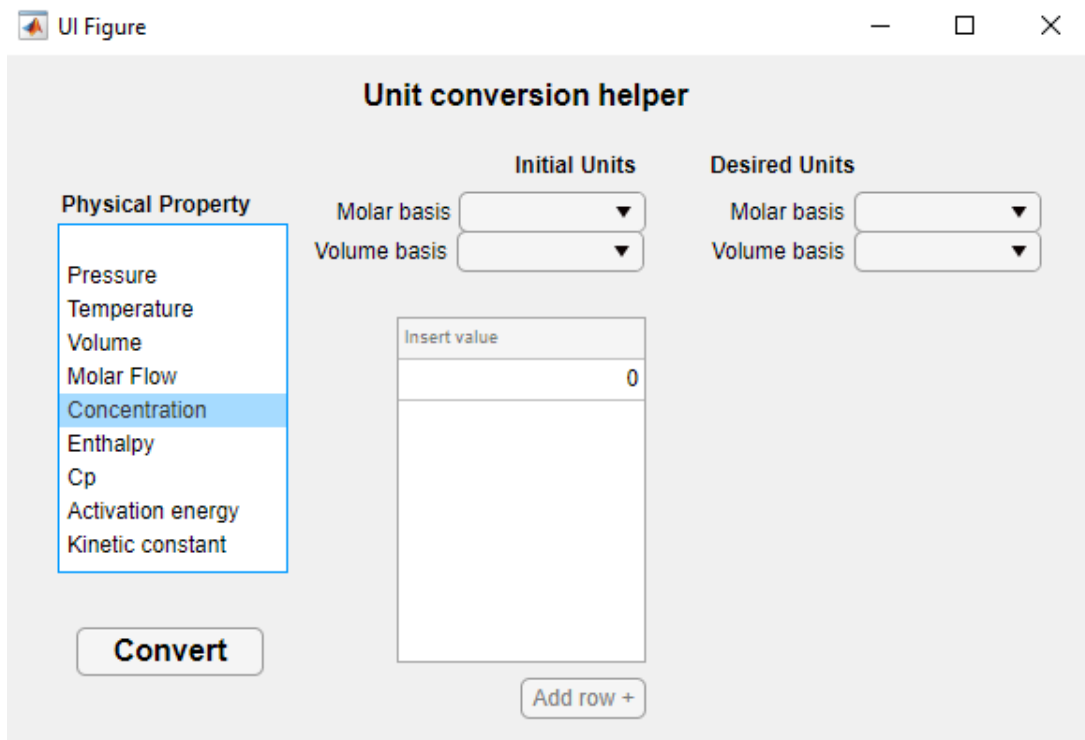


Figura 18: Aspecto de la ventana del conversor de unidades. Fuente: elaboración propia.

Para hacer uso de este apoyo, se debe proceder de la siguiente manera:

1. Seleccionar con qué tipo de propiedad se quiere realizar la conversión de unidades en la tabla de la izquierda
2. En la tabla central, proporcionar los valores de entrada y las unidades en las que están expresados.
3. Indicar a qué unidades se quiere realizar la conversión en los desplegables de la derecha.
4. Hacer clic sobre el botón *Convert* y aparecerá a la derecha de la pantalla una nueva tabla con los valores expresados en las unidades deseadas.

El botón *Add row +* se ha incluido para casos en los que se quiera cambiar un vector de una base de unidades a otra. Por ejemplo, si en un sistema con dos componentes una corriente de alimentación tiene una concentración de $[25, 74]$ mmol/cm³ y se quiere pasar a mol/m³, no sería necesario introducir cada valor de uno en uno, sino que se añadiría una fila a la tabla y se podrían transformar los dos conjuntamente.

El procedimiento anterior sería ligeramente diferente cuando se quiere hacer un cambio de unidades en la constante cinética o en el factor preexponencial de la ecuación de Arrhenius (k_0) ya que estas constantes tienen unidades diferentes según el orden de la reacción. Por este motivo, en lugar de haber un desplegable con las posibles unidades de trabajo, como ocurre en el resto de propiedades y parámetros, en este caso hay que introducir manualmente tanto las unidades

iniciales como las deseadas. La única precaución es que cualquier unidad debe ir precedida de una u seguida de un punto. Para ilustrarlo se va a tratar un pequeño ejemplo. Si para una determinada reacción el factor k_0 obtenido de la bibliografía es de $50 \text{ dm}^3/\text{gmol}/\text{h}$ y quiere convertirse a unidades del SI, la forma de expresar esta orden se muestra en la figura 19.

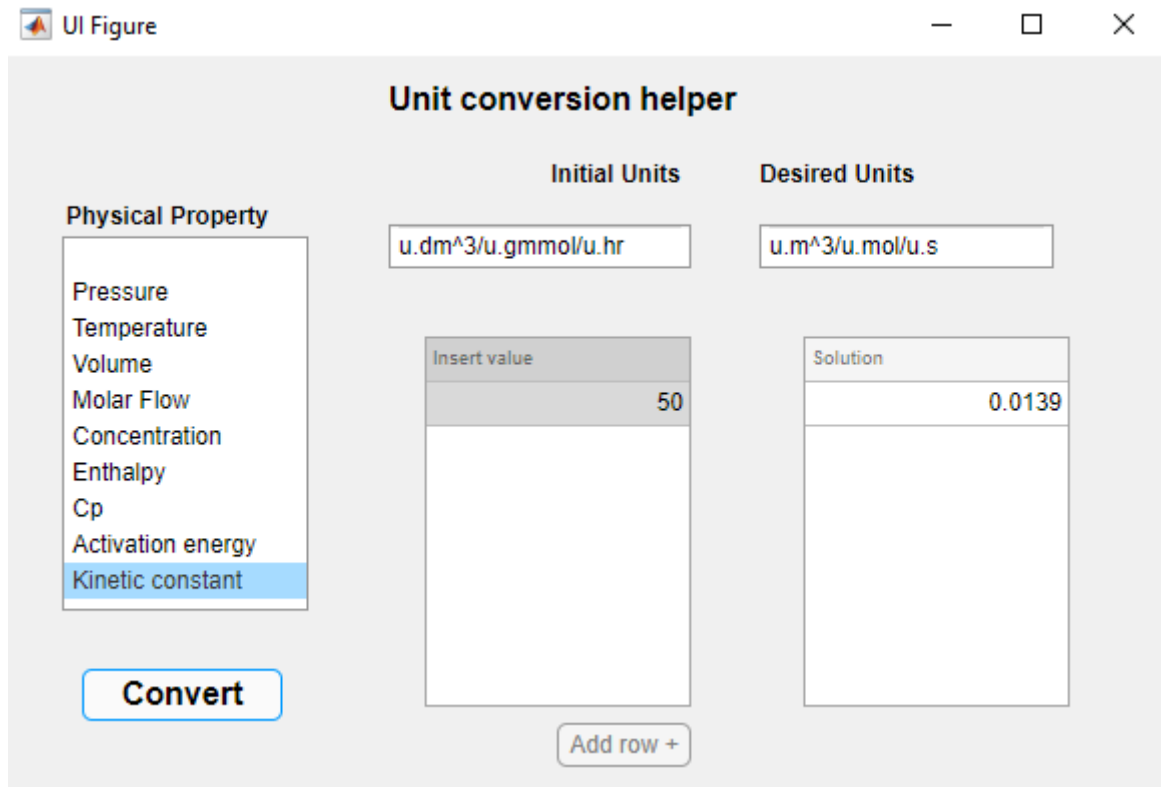


Figura 19: Ejemplo de cambio de unidades para una constante cinética. Fuente: elaboración propia.

Además del uso de “u.” resulta llamativo que las horas se expresen como “hr” o el gmol como “gmmol”. Si no se hiciera de esta manera, la aplicación no reconocería de que unidad se trata y aparecería un mensaje de error. Esto se debe a que todas las conversiones se realizan empleando el paquete de archivos *Units for Matlab* (Kitchin, 2020) por lo que es necesario adecuarse a la manera en la que este programa ha sido diseñado para que funcione de manera correcta.

Con lo expuesto hasta el momento, se ha revisado cómo pueden proporcionarse los datos del problema desde la aplicación. Esta forma de proceder, aunque resulta muy intuitiva es el paso que más tiempo consume. En el anexo I de este trabajo se han incluido ejemplos donde toda la información del problema se indica desde un script, sin necesidad de recurrir a la interfaz gráfica.

El último paso para resolver el problema consiste en volver a la ventana principal de la aplicación, introducir los nombres de las variables con las que se han guardado los datos en las casillas correspondientes, como se muestra en la figura 20, marcar si se precisa realizar una estimación económica o no y presionar el botón *Run*.

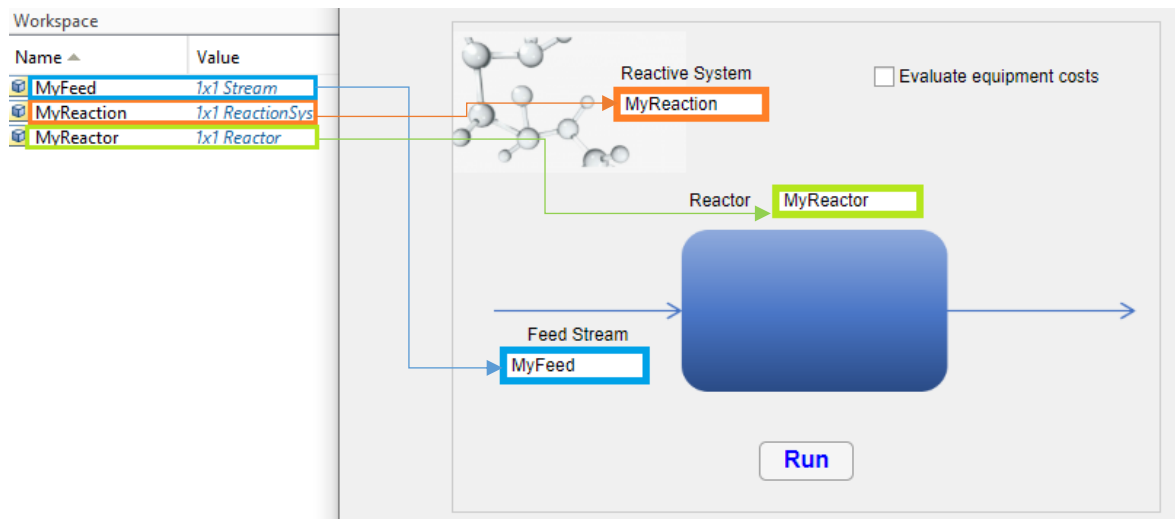


Figura 20: Plantilla de resolución de un problema desde la interfaz gráfica. Fuente: elaboración propia.

Para ilustrar los pasos presentados anteriormente, en la siguiente sección de este trabajo se explica de forma detallada cómo se ha de proceder para resolver un ejercicio de una asociación de reactores.

8.1 Ejemplo de resolución

El ejemplo se ha extraído de los problemas propuestos en la asignatura de Diseño de Reactores I y su enunciado es el siguiente (Aracil et al., 2018):

La reacción irreversible $A + B \rightarrow C$ se lleva a cabo en fase gas, a 227°C y 10 atm. La velocidad de reacción ($\text{mol}/(\text{L}\cdot\text{min})$) en función de la conversión es:

$$-r_A = 0.0167 - 0.023(X_A - 0.1) + 0.0234(X_A - 0.1)(X_A - 0.7)$$

Se va a procesar un alimento de 1 L/s que contiene 41% A, 41% B y 18% inertes (fracciones molares).

a) ¿Cuál es la conversión global si dos RCTA de 400 L cada uno se conectan en serie? (Solución: $X_A = 0,663$)

b) ¿Cuál es la conversión global si los dos reactores anteriores se conectan en paralelo, siendo el caudal volumétrico alimentado a cada uno la mitad del inicial? (Solución: $X_A = 0,578$)

Antes de introducir los datos en la aplicación, hay que adaptarlos al formato apropiado. En primer lugar, conviene darse cuenta de que este sistema no sigue el modelo de LHHW. En consecuencia, se ha proporcionado la expresión adecuada de la velocidad de reacción desde un archivo de función. Es necesario recordar que los argumentos de entrada de la función deben ser la concentración y la temperatura (en ese orden), a partir de las cuales se calcula la conversión del reactivo A tras un pequeño cálculo. El archivo se llama “kineticsExercise40.m” y su código es el siguiente:

```
function r_i = kineticsExercise40(concentration,T)

Feed = evalin('base','Feed') ;

cA = concentration(1) ;
nA0 = Feed.molarFlow(1) ;
Qv0 = Feed.volumetricFlow ;
T = Feed.T ;
P = Feed.P ;

XA = (cA*Qv0-nA0)*P/(nA0*(cA*8.314*T-P)) ;

r_i = 0.0167 - 0.023*(XA-0.1) + 0.0234*(XA-0.1)*(XA-0.7) ; %mol/L/min
r_i = r_i * 1000/60 ; %mol/m^3/s
```

Figura 21: Código de “kineticsExercise40.m” para calcular la velocidad de reacción del ejercicio 40 de DR I. Fuente: elaboración propia.

Para definir la corriente de entrada, la composición no puede introducirse como fracciones molares. Por esta razón, el último cálculo previo que habría que realizar es el caudal molar de cada especie utilizando la ley de los gases ideales. Los resultados en unidades del SI son:

$$n_{A0} = n_{B0} = 0.0999 \text{ mol/s} ; \quad n_{C0} = 0 \text{ mol/s} ; \quad n_{inertes} = 0.0438 \text{ mol/s}$$

Hecho esto, se procede a ejecutar la aplicación y seleccionar el tipo de problema adecuado, que en este ejemplo es el de asociación de reactores, y señalar que, en primer lugar, se van a conectar en serie.

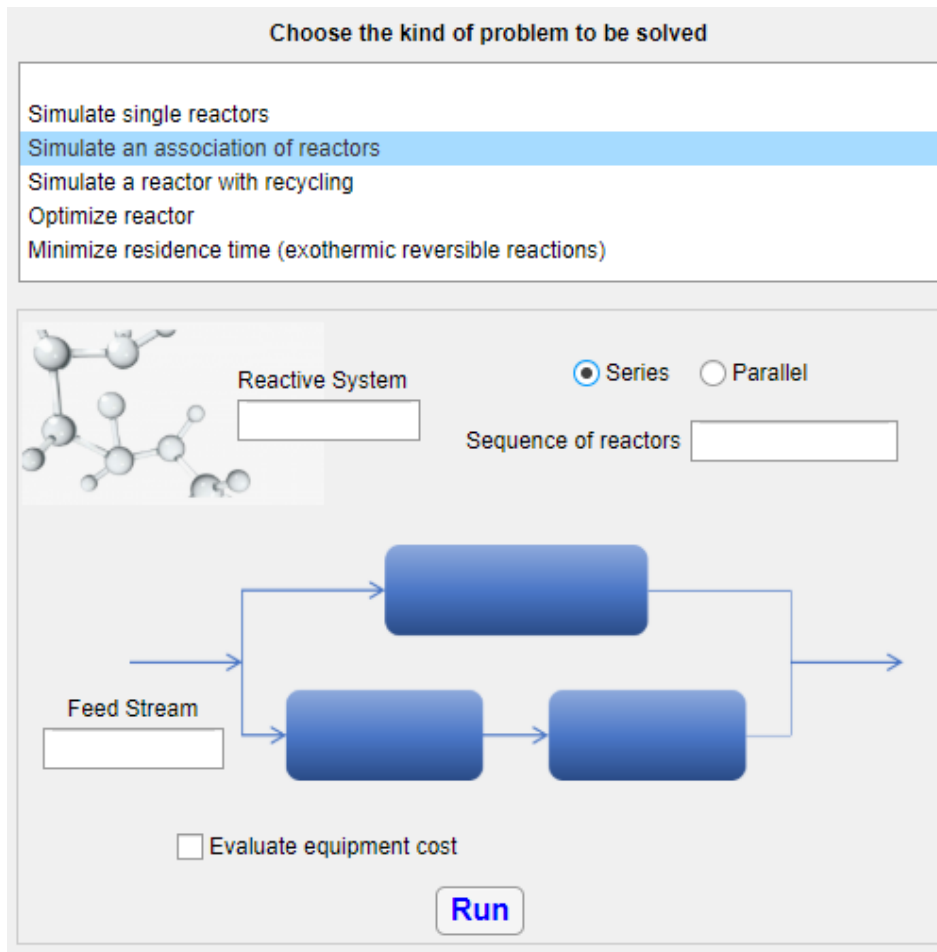


Figura 22: Selección del tipo de problema apropiado para resolver el ejemplo. Fuente: elaboración propia.

Observando la figura 22, es posible advertir qué datos deben indicarse: una corriente de alimentación, un sistema reaccionante y la red de reactores. A continuación, se introduce toda la información conocida tal y como se muestra en las figuras 23 a 26.

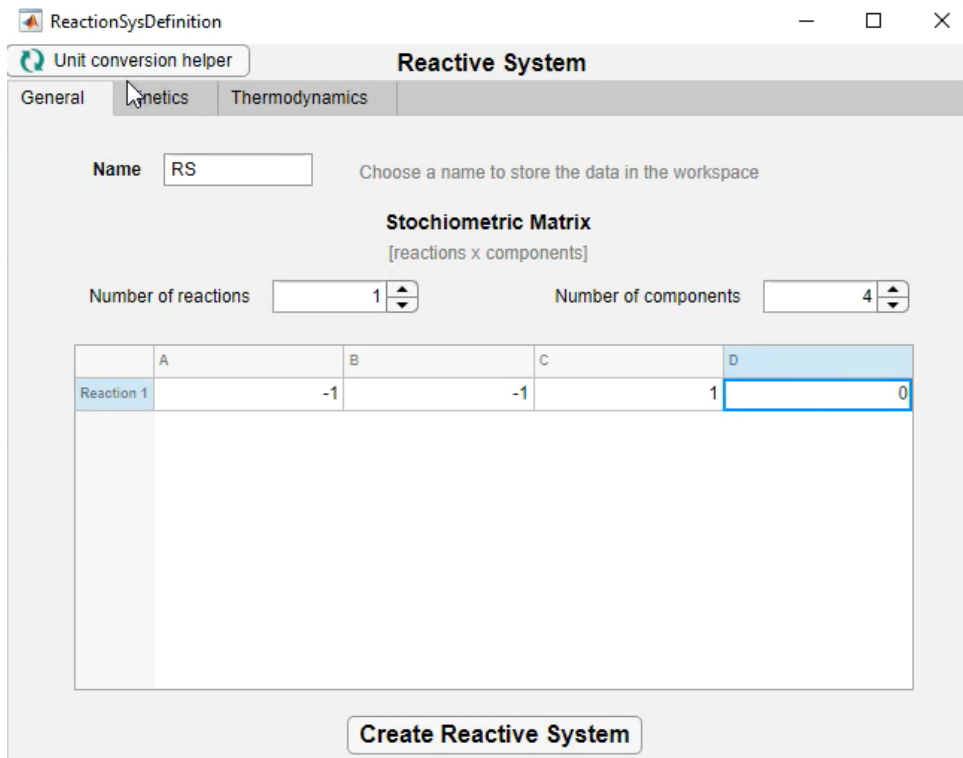


Figura 23: Introducción de la estequiometría de la reacción del ejercicio. Fuente: elaboración propia.

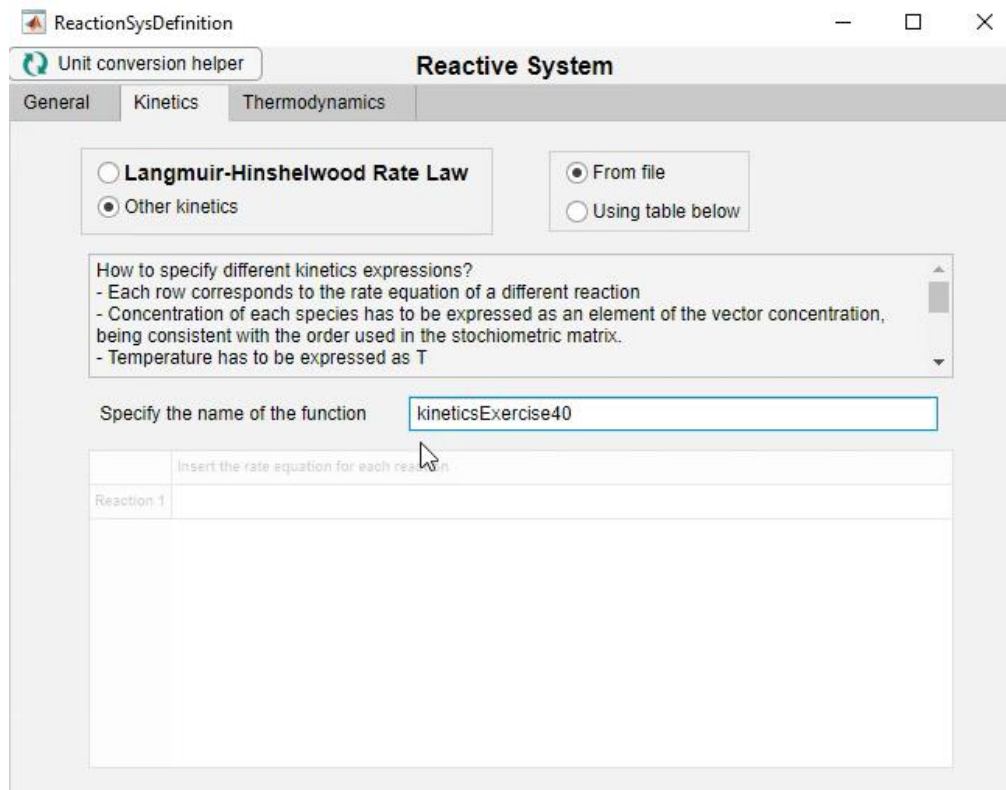


Figura 24: Especificación de la cinética de la reacción del ejercicio. Fuente: elaboración propia.

Unit conversion helper
Stream
Define stream from Hysys

TIPS
 If a PRODUCT stream is being defined and you do not want to specify :
 - P or T >> select the check box.
 - The composition of any species >> simply do not enter any data in the corresponding cell

Number of components

	Molar Flow	Concentration
Component 1	0.0999	
Component 2	0.0999	
Component 3	0	
Component 4	0.0439	

Name

Phase

P (Pa)

T (K)

Volumetric Flow (m³/s)

Density (kg/m³)

Viscosity (Pa·s)

Create Stream

Figura 25: Introducción de los datos sobre la corriente de alimento del ejercicio. Fuente: elaboración propia.

Unit conversion helper
Reactor
Display plots
off on

Name Choose a name to identify the reactor

Type of reactor

PFR
 CSTR
 Batch

Volume (m³)

Bypass Ratio

Heat exchange mode

Isothermal
 Adiabatic
 Other

Does pressure change inside the reactor?

Constant
 Non constant

Mark if the reactor is catalytic

Density (kg/m³)

Porosity

Create Reactor

Figura 26: Introducción de la información acerca del reactor del ejercicio. Fuente: elaboración propia.

Una vez hecho esto, desde la ventana principal se rellenan los campos con los nombres de las diferentes instancias que se han creado, tal y como aparece en la figura 27, con la opción de calcular el coste si fuera preciso.

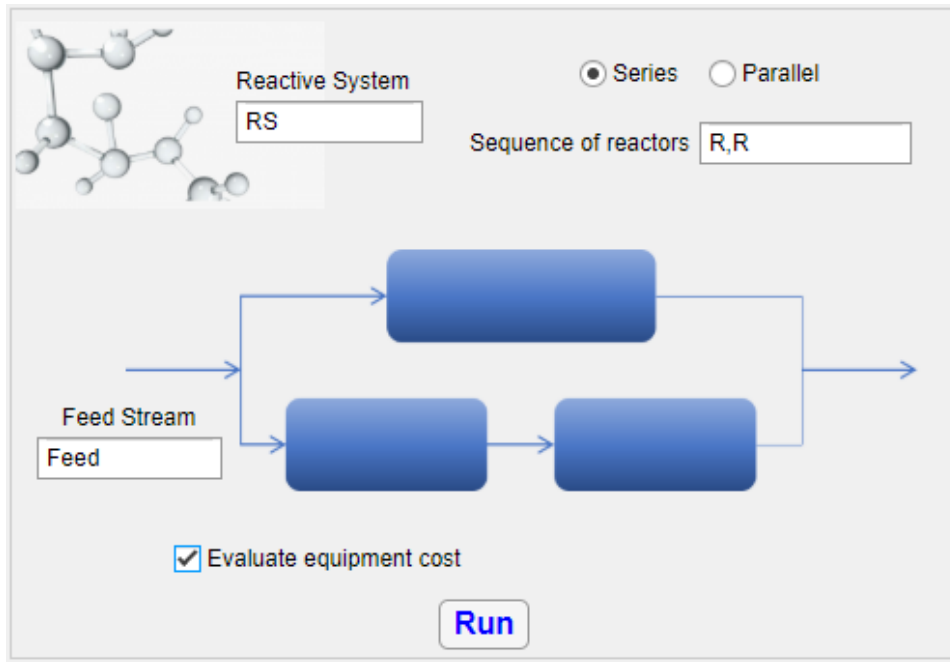


Figura 27: Formulación del problema con los reactores en serie. Fuente: elaboración propia.

Al pulsar sobre el botón *Run*, aparece una nueva casilla junto a la selección *Evaluate equipment cost* con el importe del coste total anualizado del conjunto de reactores (figura 29) y, además, se abre una nueva ventana como la de la figura 28 con los datos del efluente del sistema, a partir de los cuales se calcula la conversión global obtenida.

This window is showing the product stream which gives the solution to the given problem

Number of components

	Molar Flow	Concentration
Component 1	0.0336	46.1600
Component 2	0.0336	46.1600
Component 3	0.0663	91.0945
Component 4	0.0439	60.2580

Name: Product
 Phase: G
 P (Pa): 1.013e+06
 T (K): 500.1
 Volumetric Flow (m³/s): 0.0007279
 Density (kg/m³): 0
 Viscosity (Pa·s): 0

Figura 28: Visualización del resultado del ejercicio con los reactores en serie. Fuente: elaboración propia.

$$X_{A,serie} = \frac{0,0999 - 0,0336}{0,0999} = 0,663$$

Para resolver el segundo apartado del problema se procedería de forma idéntica y, sin necesidad de introducir los datos de nuevo, puede seleccionarse la opción de reactores en paralelo. Como puede verse en la figura 29, no es necesario indicar cómo se divide la corriente de alimento ya que por defecto se reparte equitativamente.

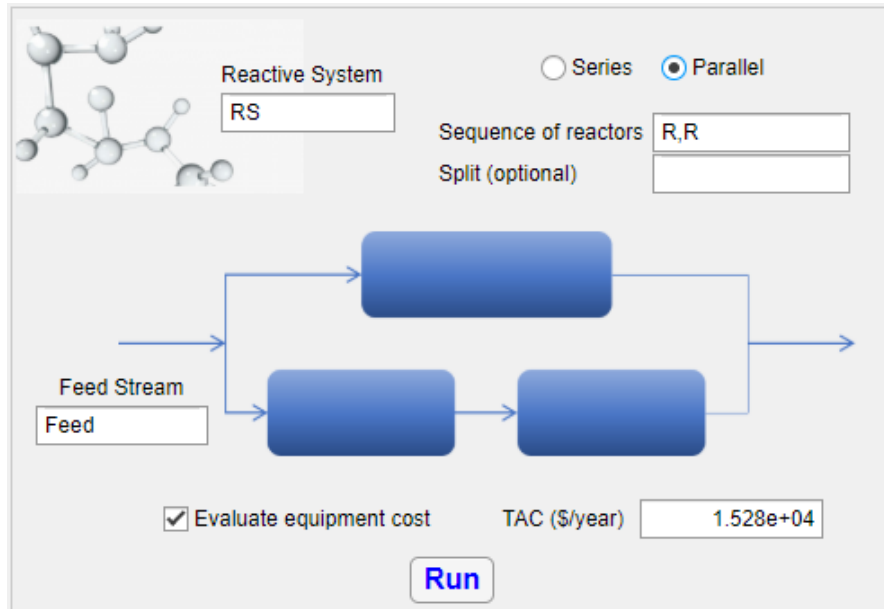


Figura 29: Formulación del problema con los reactores en paralelo. Fuente: elaboración propia.

This window is showing the product stream which gives the solution to the given problem

Number of components

	Molar Flow	Concentration
Component 1	0.0421	55.1888
Component 2	0.0421	55.1888
Component 3	0.0578	75.7912
Component 4	0.0439	57.5035

Name: Product
 Phase: G
 P (Pa): 1.013e+06
 T (K): 500.1
 Volumetric Flow (m³/s): 0.0007628
 Density (kg/m³): 0
 Viscosity (Pa·s): 0

Figura 30: Visualización del resultado del ejercicio con los reactores en paralelo. Fuente: elaboración propia.

$$X_A = \frac{0,0999 - 0,0421}{0,0999} = 0,578$$

A la vista del resultado mostrado en las figuras 28 y 30, se verifica que los cálculos son correctos.

9 DISCUSIÓN DE RESULTADOS

En el desarrollo de esta herramienta de simulación de reactores, se han llevado a cabo numerosas pruebas para tratar de detectar y corregir errores, así como garantizar que la solución proporcionada es fiable. Por este motivo, en lugar de abordar en profundidad cada uno de estos ejemplos, se ha considerado más oportuno exponer una serie de observaciones sobre la bondad de los resultados realizadas durante esta etapa de validación.

De forma general, la aplicación que se ha creado proporciona unos resultados adecuados, ya que la solución obtenida coincide con el valor esperado en gran parte de los problemas estudiados. Esta afirmación se corrobora no solo para ejemplos tan sencillos como el que se ha resuelto en el apartado anterior sino también en la simulación de sistemas más complejos que impliquen múltiples reacciones e incluso corrientes de recirculación, como puede ser el ejemplo de un RFP utilizado para la fabricación de metanol en el que tienen lugar 6 reacciones diferentes en las que participan 8 compuestos. Para ilustrarlo, la figura 31 demuestra que el perfil de temperatura en este reactor que se ha calculado con **ReactorApp** y el de la plantilla empleada para validar la solución son totalmente idénticos.

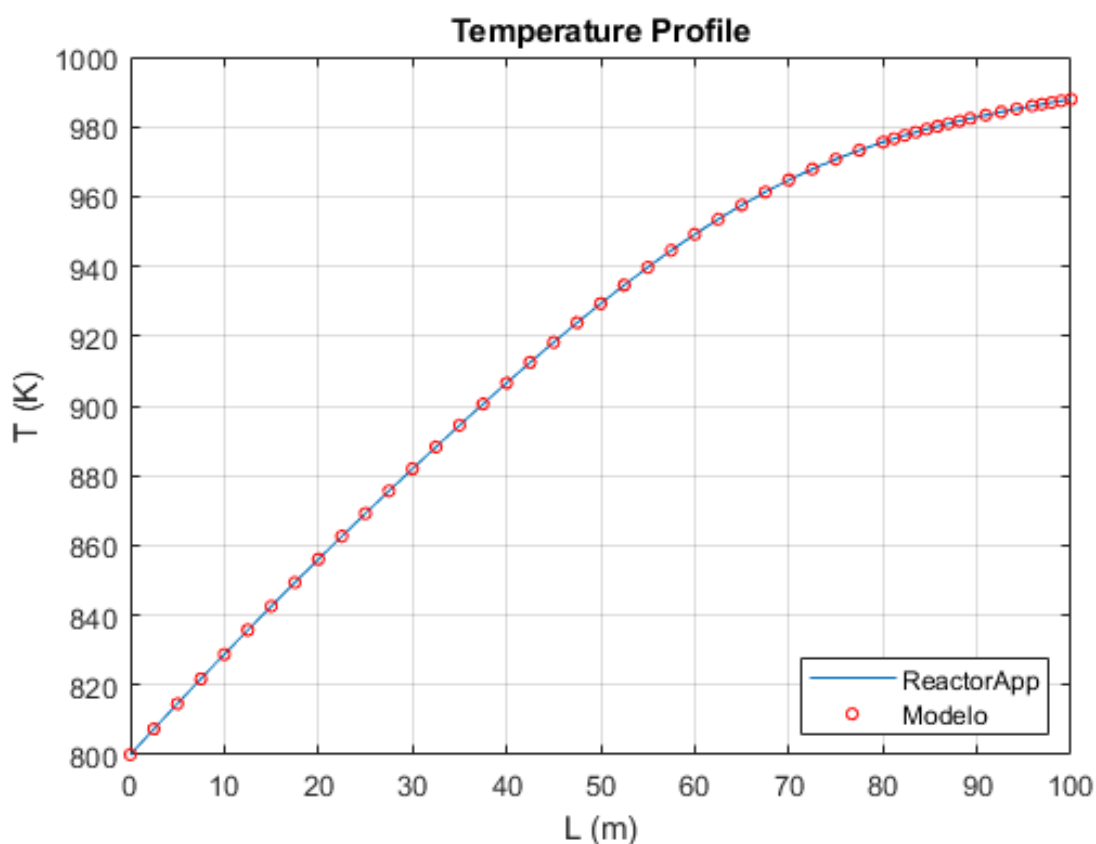


Figura 31: Comparativa del perfil de temperatura generado por la aplicación y el modelo a lo largo del RFP utilizado para la producción de metanol. Fuente: elaboración propia.

Sin embargo, no solo se ha comparado el desempeño de esta herramienta con respecto a otros ejemplos resueltos en MATLAB sino también con otros programas de simulación, concretamente Aspen HYSYS.

En el anexo I de esta memoria se ha incluido el código en el que se simula un reactor catalítico de lecho fijo utilizado en la producción de estireno, haciendo uso de la conexión con Aspen HYSYS para importar las propiedades de los compuestos presentes en el sistema. En cuanto a la obtención de las propiedades termodinámicas, es interesante resaltar que el ajuste de los puntos de calor específico en función de la temperatura a un polinomio de segundo orden es muy bueno. La función que establece la comunicación con el mencionado programa de simulación genera gráficos como el de la figura 32, en el que se representa la variación de la C_p del etilbenceno con la temperatura. Por una parte, estos gráficos permiten visualizar que la tendencia seguida es coherente, es decir, al aumentar la temperatura se debe aportar más energía para continuar calentando y, por otra, sirven comprobar que la calidad del ajuste es apropiada. En el caso del etilbenceno mostrado en la figura 32 el coeficiente de determinación R^2 es de 0,999.

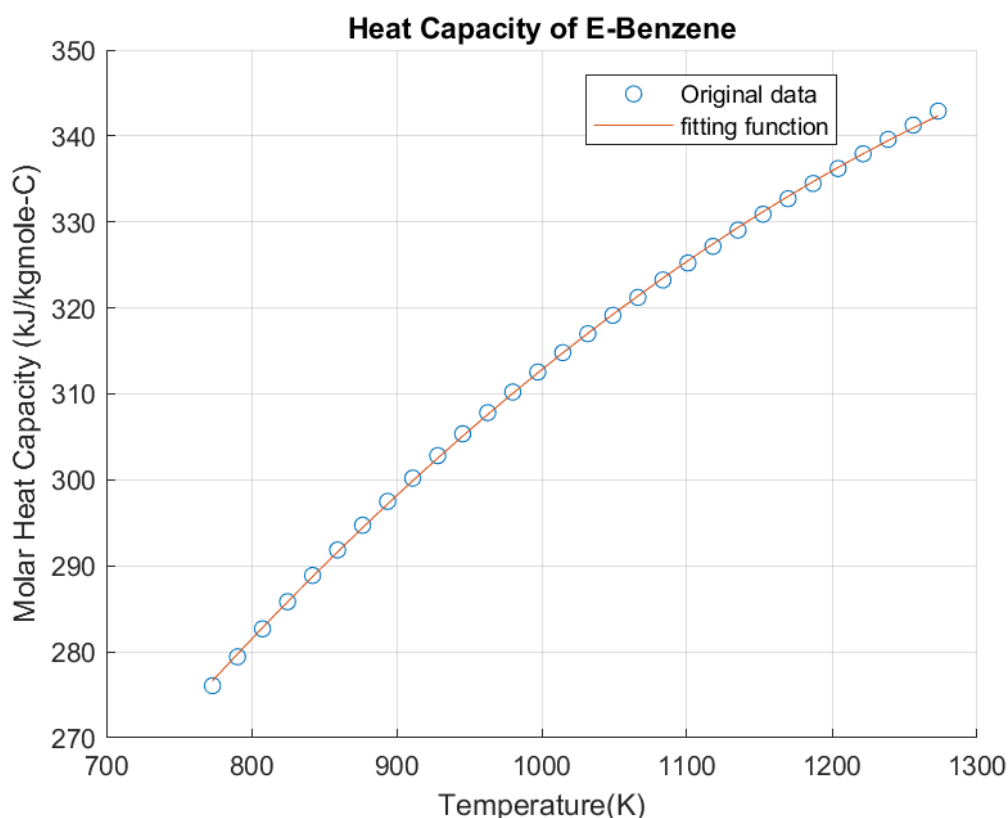


Figura 32: Ajuste del calor específico del etilbenceno a un polinomio de segundo grado. Fuente: elaboración propia.

En cuanto al resultado de la simulación, en la tabla 7 se recogen los valores de temperatura, presión y composición de la corriente de salida del reactor calculados con la aplicación

desarrollada y los que se han obtenido simulando el mismo reactor y en las mismas condiciones con Aspen HYSYS.

Tabla 7: Comparativa de la corriente de producto calculada con **ReactorApp** y Aspen HYSYS.

		ReactorApp	Aspen HYSYS	Desviación (%)
Caudal molar (kmol/h)	Etilbenceno	7,38	8,24	10%
	Estireno	12,0	12,15	1%
	Benceno	2,39	1,79	33%
	Tolueno	1,53	1,13	36%
	Hidrógeno	20,6	17,40	18%
	Etileno	1,38	1,17	18%
	Metano	0,43	0,37	16%
	Agua	269	270	1%
	Monóxido de carbono	$2,39 \cdot 10^{-2}$	$2,07 \cdot 10^{-2}$	15%
	Dióxido de carbono	2,99	1,87	60%
T (°C)		623	613	2%
P (kPa)		189	195	3%

Fuente: elaboración propia.

Como puede apreciarse, aunque los resultados son muy similares, existe una cierta desviación entre la solución generada por los dos simuladores utilizados. Estas pequeñas diferencias se atribuyen a las simplificaciones realizadas en el desarrollo de las ecuaciones utilizadas para modelar cada uno de los reactores. Por ejemplo, asumir que el gas se comporta de manera ideal y utilizar la ley de los gases ideales puede ser una fuente de error ya que el comportamiento del gas en el interior del lecho sólido puede desviarse en gran medida de la idealidad. En este sentido, Aspen HYSYS utiliza paquetes termodinámicos específicos para modelar el comportamiento de los gases de forma más realista. Además, no se ha tenido en cuenta el papel de la difusión externa e interna a través de los poros del catalizador mientras que en HYSYS sí que se tienen en cuenta aspectos relacionados con la geometría de las partículas sólidas, como el porcentaje de esfericidad, que posiblemente sean utilizados para evaluar estos efectos.

A pesar de ello, resulta muy satisfactorio haber logrado un funcionamiento semejante partiendo de que la comparación se está realizando con un software profesional. Por tanto, puede concluirse que, aunque el cálculo efectuado por las funciones que componen la aplicación de **ReactorApp** no sea tan preciso como el que proporciona Aspen HYSYS, ofrece una buena estimación de las características del producto formado y puede resultar igualmente útil cuando el objetivo sea analizar la evolución de las diferentes variables a lo largo del reactor pues, como se observa en las figuras 33, 34, 35 y 36, los perfiles de temperatura, presión y caudal molar de cada especie muestran tendencias similares.

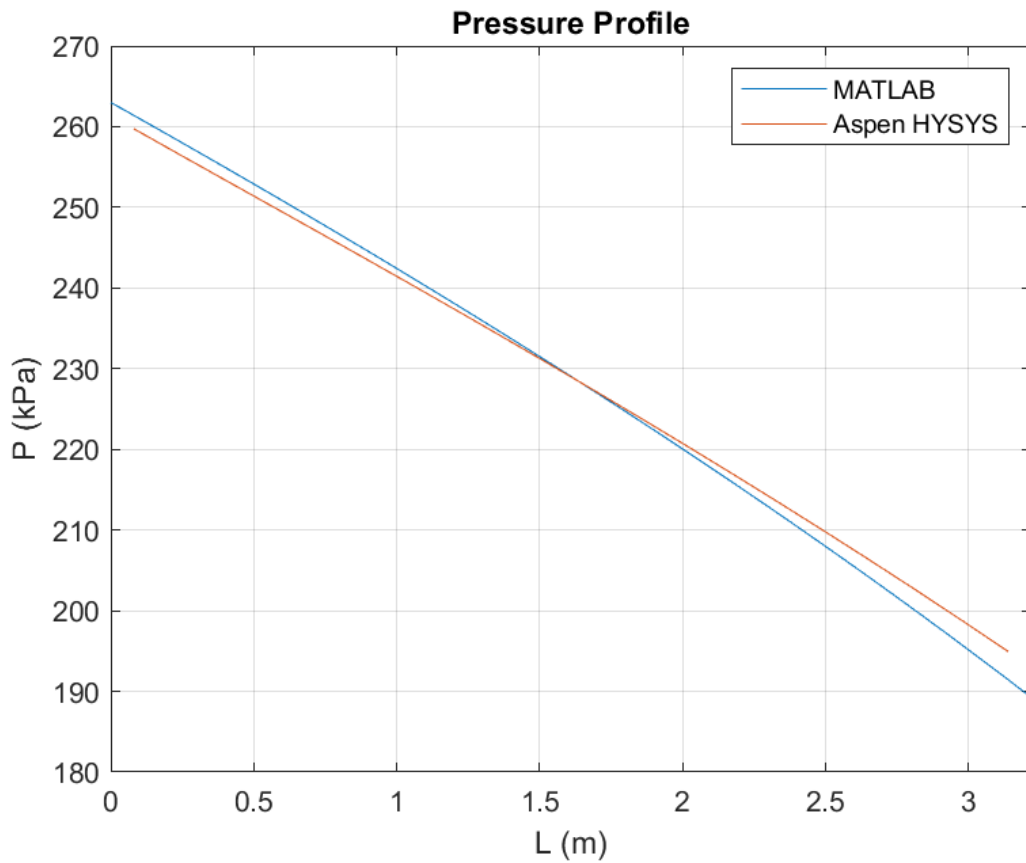


Figura 33: Evolución de la presión a lo largo del reactor obtenida con **ReactorApp** y con Aspen HYSYS. Fuente: elaboración propia.

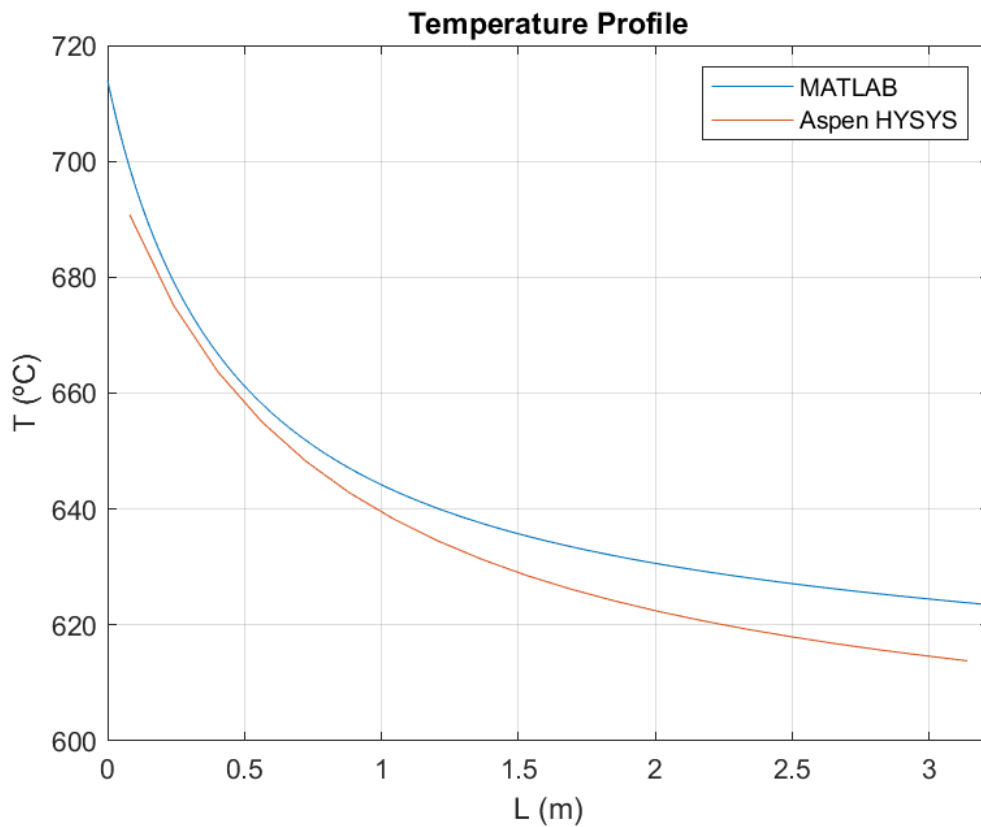


Figura 34: Perfil de temperatura generado con **ReactorApp** y con Aspen HYSYS. Fuente: elaboración propia.

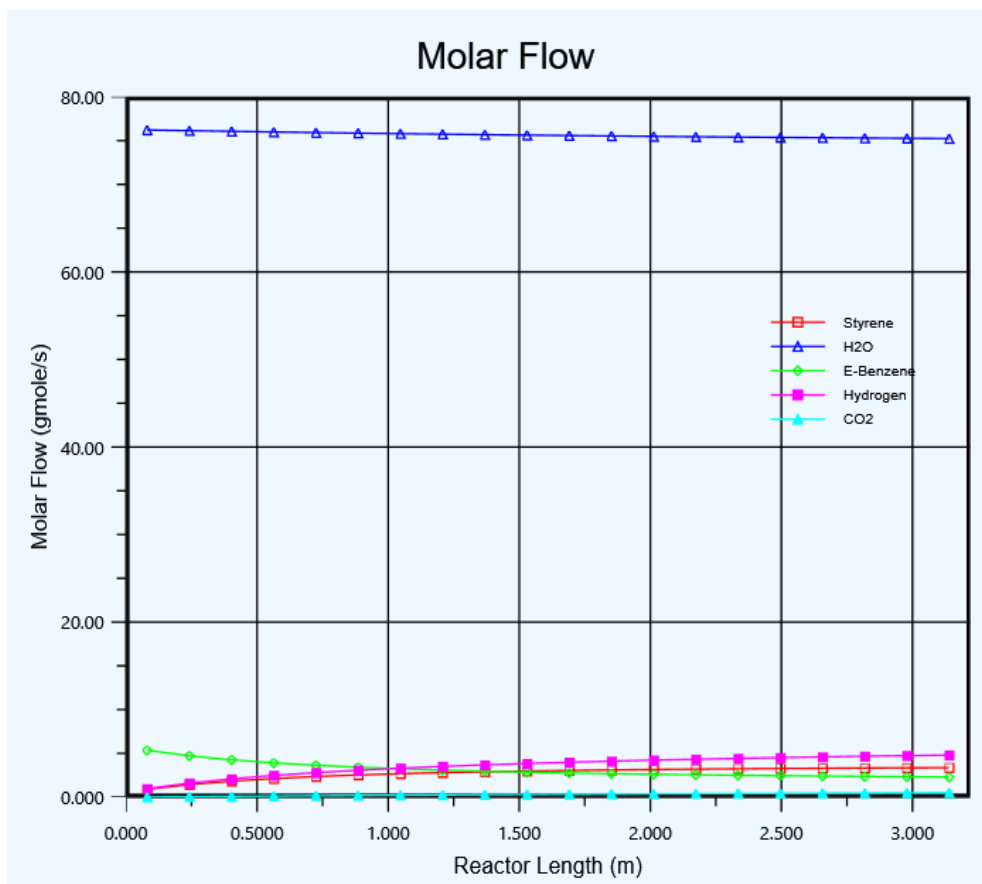
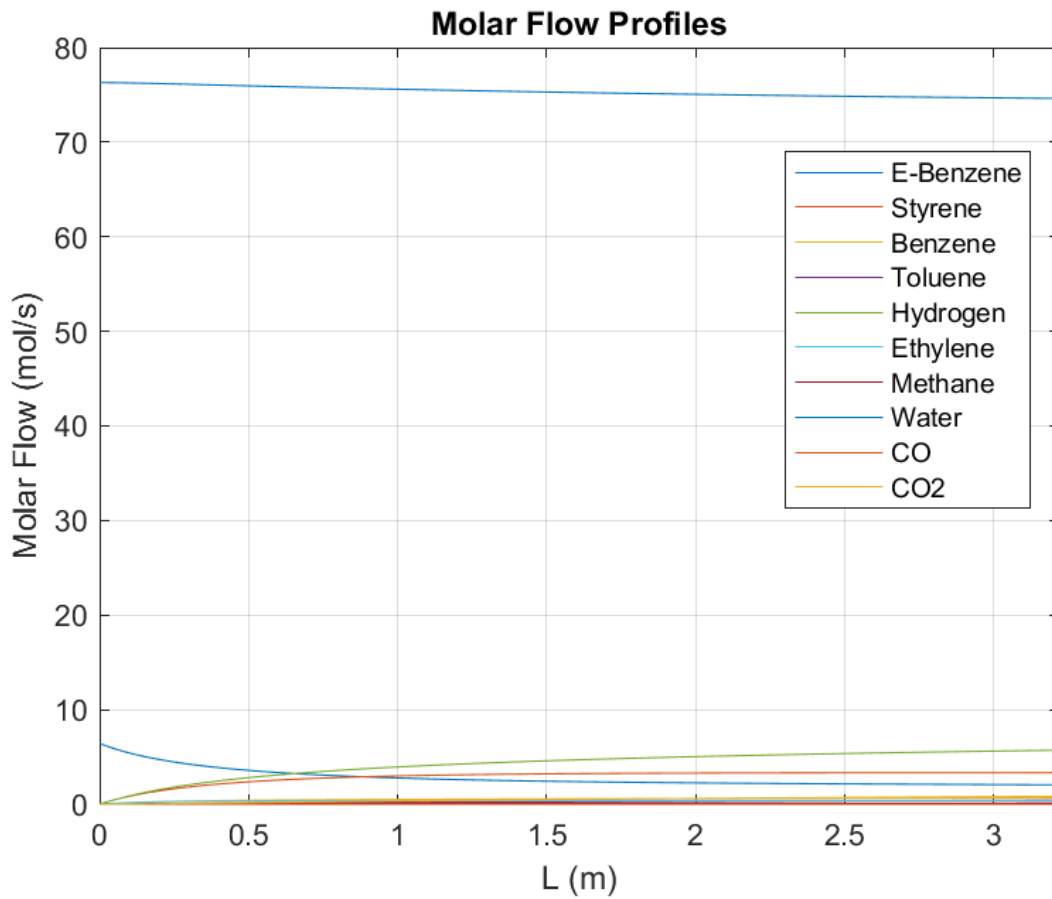


Figura 35: Evolución del caudal molar de distintos compuestos a lo largo del reactor obtenida con **ReactorApp** (arriba) y con Aspen HYSYS (abajo). Fuente: elaboración propia.

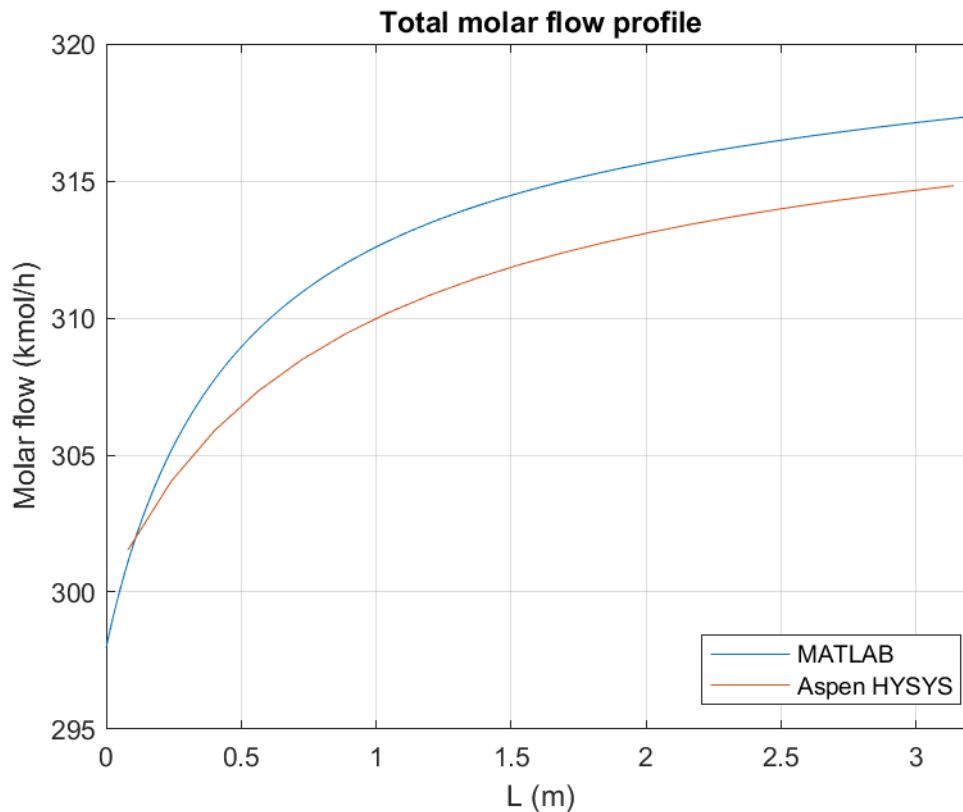


Figura 36: Evolución del caudal molar total a lo largo del reactor. Fuente: elaboración propia.

Por último, en la etapa de validación también se analizó el desempeño de las funciones de optimización descritas en los puntos 6.7 y 6.8 de este trabajo. Al igual que en los casos anteriores, se resolvieron ejemplos de diferentes problemas de optimización cuyo estudio permitió tanto detectar cuáles pueden ser las principales dificultades que se puede encontrar el usuario a la hora de interpretar la solución como identificar cuáles son los puntos fuertes de estas funciones.

Para explicar en qué consisten dichas ventajas e inconvenientes, conviene centrarse en las comprobaciones realizadas para verificar que la función que genera el perfil de temperatura óptimo trabaja correctamente. En este caso, se resolvieron los diferentes ejemplos del capítulo 9 del Levenspiel, donde se explica esta metodología, en los que se trata de optimizar un reactor de flujo pistón. El aspecto más atractivo de esta función es que es capaz de generar los diagramas de conversión frente a temperatura característicos de las reacciones exotérmicas reversibles, en los que aparecen trazadas las curvas de velocidad de reacción constante. Resulta muy útil que la aplicación genere este tipo de figuras ya que su elaboración requiere de una serie de conocimientos de optimización y de métodos computacionales que un alumno que se enfrenta a las asignaturas de diseño de reactores puede que no haya adquirido todavía. Salvada esta barrera, el apoyo de representaciones como las que elabora esta función facilita mucho el estudio de esta clase de reacciones y la comprensión de este tipo de problemas. Sin embargo, se ha comprobado que el resultado final de la optimización, es decir, de la función objetivo, difiere con respecto al

que se indica en el libro de referencia, como puede verse en la tabla 8 para el caso del ejemplo 9.4 (Levenspiel, 2004) en el que se trata de minimizar el volumen y el tiempo de residencia de un RFP que opera siguiendo la progresión de temperatura óptima.

Tabla 8: Resultados del ejemplo 9.4 del Levenspiel.

	Levenspiel	ReactorApp
Tiempo de residencia (min)	1,62	1,44
Volumen (L)	405	361

Nota. Adaptado de “*Chemical Reaction Engineering*”, de Levenspiel, O., 2004, Limusa Wiley

Las diferencias se atribuyen a que, como se ha recurrido al método de los trapecios para integrar la ecuación 32, el número de divisiones considerado afecta al resultado final. Además, los algoritmos que utiliza el comando de MATLAB *fmincon*, que es el que se ha utilizado para resolver estos problemas, son algoritmos de optimización local. Por este motivo, el valor de temperatura que se considere para iniciar el cálculo posee una gran influencia sobre la solución final obtenida, lo cual puede ser una fuente de error. Aunque desde la interfaz gráfica no puede modificarse dicho valor inicial, sí que sería posible vencer este obstáculo acudiendo al código de la función, en el cual se ha tratado de indicar de forma más clara posible la línea que debería ser modificada cambiar. En el anexo I de este trabajo se ha incluido el código para hallar el tiempo de reacción mínimo en un RDTA y se detalla cómo debería aplicarse dicha modificación. Otra de las limitaciones de esta función se encuentra a la hora de resolver problemas en los que el reactor es adiabático debido a que, en este caso, es necesario utilizar un procedimiento iterativo para conocer la temperatura óptima de entrada del alimento, lo cual requiere un tiempo de ejecución muy elevado y no es posible alcanzar una solución en un tiempo razonable. A partir de todo lo expuesto anteriormente, lo que se pretende concluir es que el resultado de la optimización debe analizarse con criterio, comprobando si tiene sentido para el problema planteado y asumiendo un cierto margen de error en el valor óptimo obtenido.

10 CONCLUSIÓN

Se ha desarrollado una aplicación capaz de resolver problemas típicos de simulación e incluso optimización de reactores químicos a los que cualquier ingeniero químico se enfrenta durante su etapa formativa. Este trabajo ha supuesto todo un reto a nivel de programación por dos motivos. Para que todos los archivos funcionen satisfactoriamente y ejecuten los cálculos de forma correcta, se ha profundizado en el lenguaje de MATLAB, aprendiendo nuevos comandos y formas de estructurar el código. Sin embargo, el aspecto más exigente ha sido la capacidad de abstracción requerida por este trabajo ya que, en lugar de estar enfocado en problema concreto, ha sido fundamental contemplar todas las alternativas y circunstancias que pudieran plantearse. Como resultado, se ha creado una herramienta con la que el usuario puede interactuar de forma intuitiva por medio de una interfaz gráfica o directamente desde el editor de MATLAB.

Además de las ventajas de esta aplicación en relación con la formación no presencial y un aprendizaje más autodidacta, una de sus características de mayor valor es la posibilidad de obtener las propiedades de diferentes compuestos químicos gracias a la comunicación que se establece con Aspen HYSYS. Una gran parte de los problemas que se resuelven en las asignaturas de diseño de reactores se formulan utilizando compuestos genéricos, por ejemplo, A y B reaccionan para formar la especie C. Este acceso a las propiedades termodinámicas permite que problemas que a priori partan de un planteamiento meramente teórico cobren un significado práctico pues dichos compuestos A, B y C podrían sustituirse por otros cualesquiera sin la limitación de carecer de los datos necesarios. Por otra parte, la estimación económica del coste de los equipos estudiados es otra forma de acercar estos problemas a un contexto más real, ofreciendo una aproximación de dicho importe.

No obstante, todavía existen numerosas vías de mejora para que el provecho obtenido de una aplicación como **ReactorApp** sea máximo. Algunas de estas propuestas son:

- Estudiar en mayor detalle cómo fluye el fluido refrigerante o medio calefactor e introducir un nuevo balance de energía para evaluar cómo se modifica su temperatura al entrar en contacto con la carcasa del reactor.
- Perfeccionar las funciones que resuelven problemas de optimización, no solo para admitir el tratamiento de reactores adiabáticos en el caso del camino de temperatura óptima, sino para adquirir una mayor flexibilidad a la hora de definir la función objetivo.
- Considerar desviaciones de la realidad en el comportamiento de los reactores, recurriendo a la distribución de tiempos de residencia.

- Integrar una aplicación de estas características a las prácticas de laboratorio, ya que el concepto de este trabajo puede servir para acercar los aspectos teóricos y prácticos en el aprendizaje del diseño de reactores.

11 REFERENCIAS

- Aracil, I., Fernández, M.J, Ruiz, R. (2018). *Colección de problemas de Diseño de Reactores I* [apuntes de clase]. Departamento de Ingeniería Química, Universidad de Alicante.
- AUROVA: Grupo de Automática, Robótica y Visión Artificial (2008). *¿Qué es Robolab?*
Disponible en: <http://www.aurova.ua.es/robolab/>
- Cristea, V.M., Imre-Lucaci, A., Nagy, Z.K. y Agachi, S.P. (2005). E-tools for Education and Research in Chemical Engineering. "POLITEHNICA" Univ. (Timișoara), 50(64), 1-2.
- Fogler, H. Scott. (2006). *Elements of Chemical Reaction Engineering* (4th edition). Prentice Hall.
- Kitchin, John. (2020). *Units for Matlab* (Versión 1.0). MATLAB Central File Exchange.
Disponible en: <https://www.mathworks.com/matlabcentral/fileexchange/36878-units-for-matlab>
- Levenspiel, Octave. (2004). *Chemical Reaction Engineering* (3rd edition). Limusa Wiley.
- Moulijn, J.A., van Leeuwen, P.W.N.M. y van Santen, R.A. (1993) Chemical Kinetics of Catalysed Reactions. In *Catalysis: An Integrated Approach to Homogeneous, Heterogeneous and Industrial Catalysis* (pp. 69-86). Elsevier. Disponible en: [https://doi.org/10.1016/S0167-2991\(08\)63807-0](https://doi.org/10.1016/S0167-2991(08)63807-0)
- Nieva García-Villaraco, Guillermo. (2016). *CHEM. Reactor Simulator: herramienta de simulación de CSTR como apoyo a la docencia* [Proyecto Fin de Carrera, Universidad Politécnica de Madrid]. Archivo Digital UPM. Disponible en: <http://oa.upm.es/42867/>
- Perry, Robert H. (1997). *Perry's Chemical Engineering Handbook* (7th edition). Mc Graw Hill.
- Plouffe, P., Macchi, A., y Roberge, D.M. (2014). From Batch to Continuous Chemical Synthesis—A Toolbox Approach. *Organic Process Research & Development*, 18 (11), 1286-1294. Disponible en: <https://doi.org/10.1021/op5001918>
- The Chemical Engineering Plant Cost Index (CEPCI) (2020)*. Chemical Engineering: Essentials for the CPI professionals. Disponible en: <https://www.chemengonline.com/pci>
- Turton, R. (2018). *Analysis, Synthesis and Design of Chemical Processes* (5th edition). Prentice Hall.

12 ANEXO I: EJEMPLOS DE USO DE REACTORAPP A PARTIR DE CÓDIGO

12.1 Simulación de un reactor tubular de lecho fijo

En este anexo se ha incluido el código necesario para simular un RFP de lecho catalítico que se emplea en la producción de estireno con el objetivo de demostrar cómo utilizar la herramienta desarrollada en este trabajo sin recurrir a la interfaz gráfica y, además, utilizar la conexión con Aspen HYSYS para obtener las propiedades termodinámicas. Los resultados que se obtienen al ejecutar el script se han presentado en el apartado de discusión de resultados.

```
clear, close, clc

%% Reactor data
Reactor = PFR ;
Reactor.heatMode = 'Adiabatic' ;
Reactor.pressureMode = 'Non constant' ;
Reactor.pressureDropEqn = 'Ergun' ;
Reactor.diameterTubes = 1.95; % m
crossSection = pi * Reactor.diameterTubes^2 / 4; % m^2
Reactor.L = 3.22 ; %m

Reactor.activatePlots = 'on' ;

% R.densityCatalyst = 2350 ; %kg/m^3 % No debe especificarse porque los
parámetros cinéticos ya vienen expresados por m^3 y no por kg de
catalizador
Reactor.porosityCatalyst = 0.445 ;
Reactor.particleDiameter = 1e-3 ; % m

%% Reactive system
RS = ReactionSys ;
RS.stoichiometricMatrix = [ -1  1  0  0  1  0  0  0  0  0  0 ;
                             1 -1  0  0 -1  0  0  0  0  0  0 ;
                             -1  0  1  0  0  1  0  0  0  0  0 ;
                             -1  0  0  1 -1  0  1  0  0  0  0 ;
                             0  0  0  0  4 -1  0 -2  2  0  0 ;
                             0  0  0  0  3  0 -1 -1  1  0  0 ;
                             0  0  0  0  1  0  0 -1 -1  1  1];

%[RxS]
           %  eb  st  B  T  H  E  M  W  C  D

RS.userDefinedKinetics =
@(concentration,T) styreneKinetics(concentration,T);

%% Thermodynamic data
workingDirectory = 'F:\TFG\Simulation Files\Connected to Hysys' ;
fileName = 'Styrene_reactor_v5' ;
streamName = 'Feed_Copy' ;
RS = RS.setHysysProperties(workingDirectory,fileName,streamName) ;
RS.componentCp.option = 'Cp = f(T)' ;

%% Feed stream
```

```

Feed = Stream ;
Feed.T = 714 + 273.15; % K
Feed.P = 263000; % Pa

FeedMolarFraction = [0.0778,0.0001,0,0.0003,0,0,0,0.9217,0,0]; % [1xS]
                    % eb st B T H E M W C D
FeedTotalMolarFlow = 298; % kmol/h
Feed.molarFlow = FeedTotalMolarFlow*FeedMolarFraction*1000/3600 ;
% [1xS] mol/s
Feed.molarFlow_Units = 'mol/s' ;
viscosity = 2.944e-2; % cP = mPa s
Feed.viscosity = viscosity / 1000 ; % Pa·s
Feed.phase = 'G' ;

%% Solution
Product = Reactor.compute_output(Feed,RS) ;
Product.molarFlow = Product.molarFlow * 3600/1000 ;
Product.molarFlow_Units = 'kgmole/h' ;

componentNames = ["EBenzene", "Styrene", "Benzene", "Toluene",...
"Hydrogen", "Ethylene","Methane","Water","CO","CO2"];
% RS.componentNames es un vector de celdas. Este formato no es válido
con la función fprintf y por eso se ha vuelto a definir el nombre de los
compuestos como un vector de caracteres
fprintf('Molar flow in the product stream of %s = %f kgmole/h
\n',[componentNames;Product.molarFlow])
fprintf('Temperature of the product stream = %f C \n',Product.T-273.15)
fprintf('Pressure of the product stream = %f kPa \n',Product.P/1000)

%% Function to compute the rate of reaction

function r_i = styreneKinetics(concentration,T)
Rg = 8.314 ; % Pa·m³/mol/K
partialPressure = concentration * Rg * T ; % Pa
partialPressure = partialPressure / 101325 ; % atm
A =[1967,1967/exp(-1.61), 7.3e8, 1748, 1209, 69.11, 7329];
% kmol/m³/atm/s (atm,atm²,atm³)
E = [90792.7,90792.7,207944.6,91462.1,103972.3,65688.73,73554.64];
%[1xR] kJ/kmol

% 1:eb 2:st 3:B 4:T 5:H 6:E 7:M 8:W 9:C 10:D
r_i = zeros(1,7) ;
r_i(1) = A(1) * exp (-E(1)/Rg/T) * partialPressure(1) ; %kmol/m³/s
r_i(2) = A(2) * exp (-E(2)/Rg/T) *
partialPressure(2)*partialPressure(5) ;
r_i(3) = A(3) * exp (-E(3)/Rg/T) * partialPressure(1) ;
r_i(4) = A(4) * exp (-E(4)/Rg/T) *
partialPressure(1)*partialPressure(5) ;
r_i(5) = A(5) * exp (-E(5)/Rg/T) *
partialPressure(6)*partialPressure(8)^2 ;
r_i(6) = A(6) * exp (-E(6)/Rg/T) *
partialPressure(7)*partialPressure(8) ;
r_i(7) = A(7) * exp (-E(7)/Rg/T) *
partialPressure(9)*partialPressure(8) ;

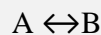
r_i = r_i * 1000 ; %mol/m³/s
end

```


12.2 Optimización de un RDTA

El ejemplo que se resuelve en este apartado pertenece al conjunto de problemas de reactores discontinuos de la asignatura de Diseño de Reactores I (Aracil et al, 2018), cuyo enunciado es el siguiente:

La temperatura en un RDTA no debe superar los 500°C mientras tiene lugar la siguiente reacción:



Si la composición del alimento es $C_{A0} = 1 \text{ kmol/m}^3$ y $C_{B0} = 0$, y se desea alcanzar una conversión del 80% de A, calcular:

- La temperatura a la cual se debe llevar a cabo la reacción para minimizar el tiempo de reacción y el valor del mismo.
- El perfil de temperatura frente a concentración de A óptimo (gráfico o tabla) para que el tiempo de reacción sea mínimo, y el valor del mismo.

Datos:

$$r_1 = 1.785 \cdot 10^5 \cdot \exp(-12000/T) C_A$$

$$r_2 = 1.172 \cdot 10^{13} \cdot \exp(-24000/T) C_B \text{ en kmol}/(\text{m}^3 \cdot \text{min})$$

Si se pretendiera abordar este problema sin hacer uso de la interfaz gráfica, debería escribirse un script con una estructura similar al código que se indica a continuación.

```
clear, close, clc

RS = ReactionSys ;
RS.stoichiometricMatrix = [-1 1 ; 1 -1] ;
k0 = [1.785e5 , 1.172e13] ; % min^-1
RS.k0 = k0 / 60 ; % s^-1
Ea_Rg = [12000 24000] ;
RS.Ea = Ea_Rg * 8.314 ; % J / mol

Feed = Stream ;
Feed.concentration = [1000 0] ; % mol/m^3
Feed.volumetricFlow = 1 ; % m^3/s
Feed.phase = 'L' ;

Product = Stream ;
desiredConversion = 0.8 ;
keyComponentIndex = 1 ;
extent = Feed.concentration(keyComponentIndex)*desiredConversion ;
Product.concentration = Feed.concentration +
RS.stoichiometricMatrix(1,:)*extent; % mol/m^3
Product.volumetricFlow = 1 ; % m^3/s
Product.phase = 'L' ;

R = Batch ;
R.heatMode = 'Isothermal' ;

minimumTime =find_optimalTemperaturePath(Feed,Product,R,...
RS,keyComponentIndex);
minimumTime = minimumTime/3600/24 ;
fprintf('The minimum reaction time when the reaction proceeds
isothermally is %.2f days',minimumTime)
```

No obstante, si se ejecutara este código directamente, la solución obtenida sería la que se muestra en la figura AI.1.

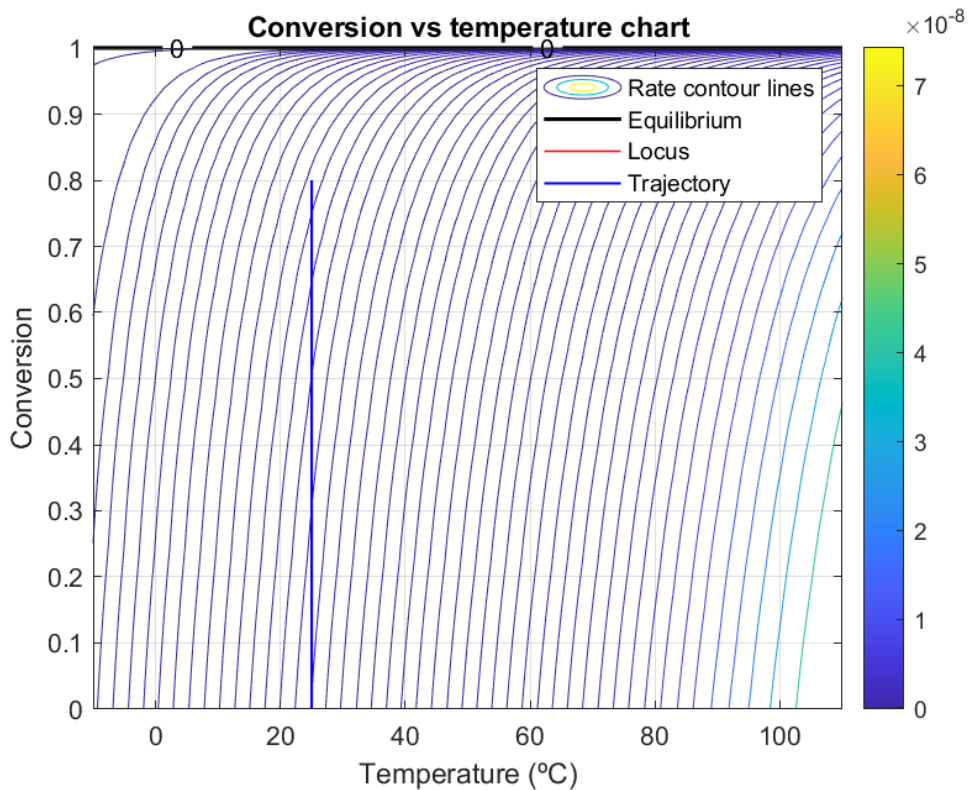


Figura AI.1: Diagrama de conversión frente a temperatura para la reacción reversible $A \leftrightarrow B$ antes de modificar la función *find_optimalTemperaturePath*. Fuente: elaboración propia.

Como puede verse, esta respuesta no es la adecuada por dos motivos. En primer lugar, en ningún momento se ha especificado que existe un límite de temperatura de 500°C por lo que el valor máximo del eje de abscisas no permite visualizar las curvas correctamente. Además, la trayectoria dibujada es errónea porque la temperatura supuesta por defecto para iniciar el algoritmo de optimización se encuentra lejos del valor óptimo. Para obtener un resultado adecuado y solventar estos inconvenientes, se debería proceder de la siguiente manera:

1. Abrir el archivo "find_optimalTemperaturePath.m".
2. Modificar la línea 27 de la función e indicar que el límite superior para la temperatura. En este caso, como son 500°C debería introducirse dicha temperatura en grados Kelvin.
3. Ejecutar el programa de nuevo para observar la forma de las curvas y determinar de forma aproximada en qué rango de temperatura se encuentra el máximo de las curvas de nivel de función de velocidad de reacción.
4. Modificar las líneas 75 y 140 de tal modo que el valor de temperatura que se tome para arrancar la optimización (x_0) esté más próximo al máximo de la curva. Por ejemplo, puede escogerse una temperatura de 600 K.

5. Volver a ejecutar la función.

Tras realizar los cambios propuestos, se obtiene una representación como la que se muestra en la figura AI.2.

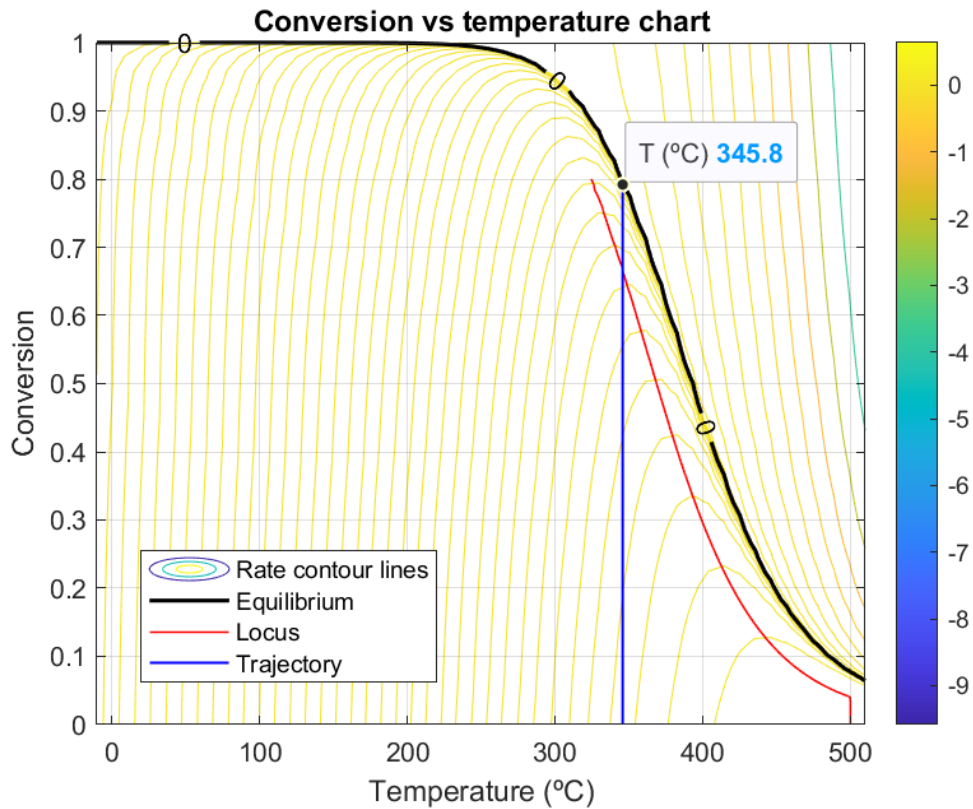


Figura AI.2: Diagrama de conversión frente a temperatura para la reacción $A \leftrightarrow B$ tras adaptar la función "find_optimalTemperaturePath.m". Fuente: elaboración propia.

A partir de la figura anterior pueden contestarse a las diferentes cuestiones planteadas. La temperatura óptima a la cual se debe llevar a cabo el proceso para minimizar el tiempo de residencia viene determinada por la recta azul mientras que el perfil de temperatura óptimo está trazado en rojo. El valor del tiempo de reacción mínimo aparecerá en la ventana de comandos de MATLAB en caso de haber ejecutado el script. Los tiempos calculados en las condiciones de ambos apartados, así como su comparación con respecto a la solución esperada se muestran en la tabla AI.1.

Tabla AI.1: Resultados de la optimización del RDTA

	t esperado (días)	t calculado (días)	Error
a) Condiciones isotermas	3,42	3,82	11%
b) Condiciones no isotermas	2,44	2,44	0,18%

13 ANEXO II: ARCHIVOS

En este anexo se han incrustado todos los archivos que componen la aplicación y que son necesarios para su correcto funcionamiento. Para ello, se han clasificado en tres categorías. En primer lugar, todas aquellas funciones o script de MATLAB utilizados para ejecutar los cálculos se han recogido en la tabla AII.1. Por otra parte, los archivos que componen la interfaz gráfica, cuya extensión es “.mlapp”, se encuentran en la tabla AII.2. Finalmente, la tabla AII.3 presenta un conjunto de archivos complementarios como imágenes o tablas a partir de los cuales se obtiene información o se ha mejorado la apariencia de la interfaz.

Tabla AII.1: Funciones de MATLAB










Archivo	Descripción
 Reactor.m	Archivo que define la clase <i>Reactor</i> .
 PFR.m	Archivos que definen las subclases de <i>RFP</i> . <i>RCTA</i> y <i>RDTA</i> Heredan los atributos y funciones de la clase <i>Reactor</i> .
 CSTR.m	
 Batch.m	
 ReactionSys.m	Archivo que define la clase <i>ReactionSys</i> (sistema reaccionante).
 Stream.m	Archivo que define la clase <i>Stream</i> (corriente).
 call_DataBase.m	Conexión con Aspen HYSYS para importar y procesar las propiedades termodinámicas de los compuestos.
 computeCost.m	Estimación económica del coste de los diferentes reactores.
 find_optimalTemperaturePath.m	Optimización de reactores siguiendo el perfil de temperatura óptima en reacciones exotérmicas reversibles.

Tabla AII.2: Archivos de la interfaz gráfica de AppDesinger (extensión ".mlapp")


















Archivo	Descripción
 ReactorApp.mlapp	Ventana principal de la aplicación.
 defineReactionSysApp.mlapp	Ventana para introducir los datos del sistema reaccionante.
 defineReactorApp.mlapp	Ventana para introducir los datos del reactor.
 defineStreamApp.mlapp	Ventana para introducir los datos de las corrientes o para mostrar el resultado, es decir, las propiedades de la corriente producto.
 dialogHysysStreamApp.mlapp	Cuadro de diálogo para indicar la localización de la simulación de Aspen HYSYS desde la que se quiere copiar una corriente.
 unitConverterApp.mlapp	Ventana del convertidor de unidades.
 UtilitiesApp.mlapp	Cuadro de diálogo para seleccionar los servicios fríos o calientes.

Tabla AII.3: Otros archivos necesarios

Archivo	Descripción
 DBIcon.png	
 HysysLogo.png	
 ImageAsocReactors.png	
 ImageMolecule.png	Conjunto de imágenes e iconos que aparecen en las diferentes pantallas de la interfaz gráfica.
 ImageRecycleReactor.png	
 ImageSingleReactor.png	
 RateHelp.png	
 UnitsLogo.png	
 UtilitiesDataBase.mat	Tabla desde la que el archivo “UtilitiesApp.mlapp” importa los datos.
 +cmu.zip	Conjunto de funciones utilizadas por el convertor de unidades. Debe descomprimirse la carpeta para asegurar un funcionamiento correcto.

Finalmente, para facilitar la distribución de la aplicación se han empaquetado todos los archivos utilizando el compilador de MATLAB, que crea el archivo “MyAppInstaller_web.exe” que, al ejecutarlo, instala la aplicación **ReactorApp**, que se ejecuta automáticamente con MATLAB Runtime. De igual manera, también están disponibles como una *toolbox* cuya descarga puede realizarse desde el portal de intercambio de archivos de MathWorks a través del siguiente enlace:

<https://es.mathworks.com/matlabcentral/fileexchange/76917-reactorapp>