

Apéndice B

Ejemplos de Solución con MathCad

A continuación vamos a ver algunos ejemplos de cómo resolver con MathCad el tipo de problemas numéricos que se presentan en este texto. Estos ejemplos corresponden a la primera vez que un método es aplicable durante el desarrollo de este texto. Se trata tan sólo de una pequeña introducción y sólo se incluye un breve caso para cada uno de los métodos que requerimos. Existen otras funciones y métodos por lo que se recomienda al alumno consultar tanto su manual de usuario como la *ayuda* del paquete. Aunque muchos ejercicios y problemas se pueden resolver directamente con estos métodos numéricos, se recomienda al alumno atacarlos utilizando el álgebra y saberlos resolver también con los métodos numéricos *básicos* que incluyen las calculadoras *científicas*: raíz de una función, integración numérica y Runge-Kutta. Debe notarse que en las soluciones incluidas para los ejemplos, cuando es posible, hemos optado por recurrir al álgebra para simplificar y poder emplear preferentemente métodos básicos obteniendo la “solución con calculadora”.

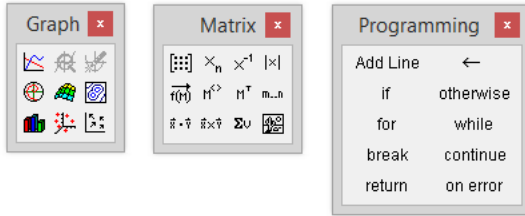
Al escribir en una hoja de MathCad es importante tomar en cuenta que la solución cualquier fórmula o definición debe estar antes y arriba. Esto es, la secuencia de solución es de *izquierda a derecha* y de *arriba a abajo*. Si alguna variable no está definida o hay algún problema de cálculo, este paquete la resalta en rojo. Cuando se redefine alguna variable, como simple advertencia se resalta en verde.

También es importante notar que existen distintos tipos de “igual”:

- := Corresponde a una definición y aparece al teclear “:=”
- = Se utiliza para hacer que el paquete que escriba algún resultado
- = Implica una definición *especial* y se obtiene de la Barra Booleana.
Por ejemplo, al resolver ecuaciones no-lineales simultáneas.

El uso de este paquete es muy intuitivo y las barras de herramientas son fáciles de interpretar y utilizar. Se recomienda evitar que MathCad maneje directamente las unidades, pero es conveniente incluirlas como *texto* o *letrero* justo enseguida de la variable. El texto se distingue porque emplea una fuente (tipo de letra) diferente. Otra sugerencia que puede aliviar confusiones es solamente usar subíndices cuando realmente se necesiten. MathCad permite que aparezcan subíndices si al teclear una variable se presiona “.”, entonces lo que se escriba a continuación se visualizará como un “subíndice”. Estos “subíndices” no son *reales* y solamente tienen un efecto en la apariencia. Se le recomienda solamente emplear subíndices *reales* generados a partir del comando “ x_n ” de la Barra de Matrices.

En este apéndice nos referiremos a distintas Barras aparecen desde la pestaña de “Vista/Herramientas”:



Generación de Gráficos

De la Barra de Gráficos, al presionar al primer botón aparece un gráfico X-Y con espacios para escribir las variables. Simplemente se teclean los nombres de las variables y listo. De manera sencilla se pueden ajustar los límites de los ejes haciendo algunos "clicks" de manera intuitiva y cambiando valores. Si deseamos más de una "Y", después de introducir el nombre de la primera variable, tecleamos "comas" y aparecen espacios adicionales. Además, con un "doble-click" directo sobre la caja interna con los ejes aparecen las "Opciones para Formatear el Gráfico X-Y Existente" con varias pestañas. La primera pestaña, Ejes X-Y, incluye un recuadro que debemos activar si necesitamos doble eje Y. El lector debe explorar las diferentes pestañas para adecuar color de curvas, símbolos para los puntos, escala normal o logarítmica, etc.

Raíces de Una Ecuación

Vamos a usar como ejemplo la Ecuación A^m en la página 81. Primero definimos algunas constantes y funciones:

$$CA0 := 0.06 \quad CB0 := 0.075 \quad CC0 := 0.0 \quad CD0 := 0.001 \quad M$$

$$KC1 := 6.436 \quad A_{dim}$$

$$CA(fB) := CA0 - \frac{2}{3} \cdot CB0 \cdot fB \quad CB(fB) := CB0 \cdot (1 - fB)$$

$$CC(fB) := CC0 + \frac{4}{3} \cdot CB0 \cdot fB \quad CD(fB) := CD0 + \frac{1}{3} \cdot CB0 \cdot fB$$

Ahora antes de invocar a la función `root`, debemos definir la función implícita, preferentemente *modificada*, y especificar una buena aproximación inicial para fB que marcamos con una x dentro del nombre para distinguirla. El resultado lo asignamos a fB_{eq} con el que podemos continuar otros cálculos. A continuación sólo se muestran para la concentración de A en el equilibrio.

$$F1m(fB) := (CC(fB)^4 \cdot CD(fB) - KC1 \cdot CA(fB)^2 \cdot CB(fB)^3) \cdot 10^0$$

$$fB_x := 0.6 \quad \text{Aproximación inicial marcada con } x$$

$$fB_{eq} := \text{root}(F1m(fB_x), fB_x) \quad \boxed{fB_{eq} = 0.585}$$

$$CA_{eq} := CA(fB_{eq}) \quad CA_{eq} = 0.0307 \quad M$$

La tolerancia para la convergencia la podemos modificar simplemente escribiendo nuestro valor antes de invocar a `root`, por ejemplo, escribimos $TOL := 0.00001$. Pero en este texto recomendamos directamente manipularla a través del factor " $\times 10^0$ ".

Evaluar Funciones Repetidamente y Exportar

Si queremos evaluar una función varias veces, exportar los datos y resultados, o incluso simplemente realizar un gráfica con cierto control por nuestra parte. Siguiendo con los datos de la sección anterior. Ahora deseamos evaluar la función $F1m$ para distintos valores de fB para exportar datos y resultados, por ejemplo, a *Excel*. Una manera de hacer es empleando la Barra de Programación. Recordando que MathCad numera los renglones y columnas de sus matrices y vectores a partir de 0, iniciamos el subíndice ind con un valor de 0 y hacemos el ciclo for para variar fB de 0 a 1. Al valor de fB lo asignamos a los elementos de la columna 0 mientras que con este valor evaluamos la función $F1m$ y el resultado lo asignamos a la columna 1. Cerramos el ciclo aumentando el subíndice ind . La última variable $Tabla$ que aparece al final es la que va a transferir los cálculos internos a la variable definida con ":", notamos que en este caso hemos empleado el mismo nombre sin problema. Como resultado, $Tabla$ es una matriz que podemos explorar para buscar cuándo la función cruza por 0 o exportar su valores. Haciendo *click derecho* con el ratón directamente sobre la tabla en la hoja de cálculo, aparece la opción de *Exportar*.

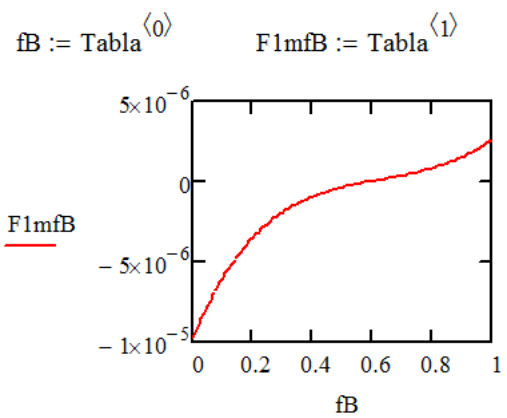
```

Tabla := | ind ← 0
         | for fB ∈ 0,0.01..1
         |   | Tablaind,0 ← fB
         |   | Tablaind,1 ← F1m(fB)
         |   | ind ← ind + 1
         | Tabla
    
```

Tabla =

	0	1
56	0.56	$-8.932 \cdot 10^{-8}$
57	0.57	$-5.322 \cdot 10^{-8}$
58	0.58	$-1.791 \cdot 10^{-8}$
59	0.59	$1.677 \cdot 10^{-8}$
60	0.6	...

También es posible asignar los valores de columnas individuales a nuevas variables para generar gráficos con sean más sencillos de interpretar. Por ejemplo, es probable que sea más fácil imaginarse qué es $F1mfB$ en un gráfico a si en su lugar apareciera $Tabla^{<0>}$.



Ecuaciones Lineales Simultáneas

Para el material cubierto en este libro, la solución numérica de ecuaciones lineales simultáneas se emplea principalmente en la estimación de parámetros por el método diferencial, siempre y cuando la expresión de velocidad se linealizale respecto a los parámetros. Consideremos un problema arbitrario de ajuste lineal de parámetros para el modelo:

$$y = \alpha_0 x_1 + \alpha_1 x_1 x_2$$

De acuerdo a lo presentado en el Apéndice D debemos obtener las sumatorias necesarias para completar los elementos del siguiente sistema:

$$\begin{bmatrix} \sum x_1^2 & \sum x_1^2 x_2 \\ \sum x_1 x_2 & \sum (x_1 x_2)^2 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} \sum x_1 y \\ \sum x_1 x_2 y \end{bmatrix}$$

En forma matricial lo podemos representar como:

$$\vec{X} \cdot \vec{\alpha} = \vec{Y}$$

El despeje para obtener el vector de parámetros $\vec{\alpha}$ no puede ser directo como en álgebra básica, para calcularlo es necesario invertir la matriz \vec{X} y luego hacer su producto-punto por el vector \vec{Y}

$$\vec{\alpha} = \left[\vec{X} \right]^{-1} \cdot \vec{Y}$$

Estos problemas los podemos resolver convenientemente en Excel porque, además de ser muy sencillas las operaciones para obtener las sumatorias, este paquete también permite con invertir y multiplicar matrices. Por otro lado, podríamos optar por realizar las sumatorias en Excel e importar esos resultados a MathCad donde se facilitan las operaciones matriciales. A continuación se presentan los puntos clave para realizar todos los cálculos en MathCad. La dificultad mayor es calcular las sumatorias definiendo variables con la Barra de Programación. Consideremos que se conocen datos y que ya se incluyeron en la matriz A donde las columnas 0, 1 y 2 corresponden a x_1 , x_2 y y , respectivamente. Para dos de la sumatorias necesarias los cálculos son:

$$\Sigma x_1 x_2 := \begin{cases} S \leftarrow 0 \\ \text{for ind} \in 0, 1 \dots 10 \\ S \leftarrow S + (A_{\text{ind}, 1})^2 \cdot A_{\text{ind}, 1} \\ S \end{cases} \quad \Sigma x_1 x_2 y := \begin{cases} S \leftarrow 0 \\ \text{for ind} \in 0, 1 \dots 10 \\ S \leftarrow S + A_{\text{ind}, 0} \cdot A_{\text{ind}, 1} \cdot A_{\text{ind}, 2} \\ S \end{cases}$$

Notamos que en la variable $\Sigma x_1 x_2$, la letra griega sigma mayúscula es parte del nombre de la variable y no una operación real de sumatoria. En las líneas de programación notamos que la variable S se usó internamente para la sumatoria y que en la última línea ésta es la variable que finalmente se regresa como resultado a $\Sigma x_1 x_2$. Como están estos códigos, esta variable tiene solamente inferencia interna dentro de las líneas de programación y pero eso no afecta que la misma variable S también se use dentro de los cálculos internos de $\Sigma x_1 x_2 y$.

Después de realizar el resto de las sumatorias se acomodan en una matriz y un vector. Se realizan las operaciones de invertir y el producto punto con la Barra de Matrices, y se asignan los resultados a la variable α .

$$X := \begin{pmatrix} \Sigma x_1 & \Sigma x_1 x_2 \\ \Sigma x_1 x_2 & \Sigma x_1 x_2^2 \end{pmatrix} \quad Y := \begin{pmatrix} \Sigma x_1 y \\ \Sigma x_1 x_2 y \end{pmatrix}$$

$$\alpha := X^{-1} \cdot Y \quad \alpha = \begin{pmatrix} 0.130 \\ 1.833 \end{pmatrix}$$

Para entender comprender esta explicación es necesario estudiar el Apéndice D en conjunto con el Capítulo 5.

Ecuaciones No-Lineales Simultáneas

Comúnmente estos sistemas aparecen cuando resolvemos problemas de equilibrio con más de una reacción independiente o reactores de tanque agitado con varias reacciones y/o varios reactores. Se recomienda siempre *modificar* las ecuaciones implícitas para evitar singularidades durante la búsqueda. Tomamos como base las dos ecuaciones no-lineales simultáneas del Ejemplo 2.9 que aparecen en la página 82. Definimos algunas variables y constantes, y las funciones para calcular los flujos dependientes a partir de los independientes:

$$\begin{aligned}
 &FA_0 := 10 \text{ Kmoles/min} \quad KP_1 := 55.108 \text{ adim} \quad PT := 2 \text{ bar} \\
 &FB_0 := 12 \quad KP_2 := 7.424 \text{ bar}^2 \quad \text{Nota: las unidades son simples letreros} \\
 &\hspace{15em} \text{que se distinguen por una fuente diferente} \\
 &FC(FA, FB) := 37 + \frac{FA}{2} - \frac{7 \cdot FB}{2} \quad FF(FA, FB) := 3 \cdot FB - FA - 26 \\
 &FD(FA, FB) := 50 + FA - 5 \cdot FB \quad FT(FA, FB) := 48 + FA - 3 \cdot FB \\
 &FE(FA, FB) := \frac{3}{2} \cdot FB - \frac{FA}{2} - 13
 \end{aligned}$$

Especificamos preferentemente buenas aproximaciones iniciales de las variables que queremos resolver. En la hoja de cálculo se incluye una *x* como parte del nombre de la variable para distinguirlas de los resultados finales. Escribimos el comando *Given* seguido por las ecuaciones implícitas a resolver. Ésta es una instrucción necesaria y no debe aparecer como simple texto. Es muy importante que el *igual* que aparece esté en negritas y se obtenga de la *Barra de Herramientas Booleana*, es decir, no es el igual empleado para definir variables ni tampoco el empleado para obtener resultados. Llamamos a la función *Find* y en ésta indicamos las variables para las cuáles debemos resolver las ecuaciones anteriores. El resto de las variables y constantes deben estar previamente definidas.

$$\begin{aligned}
 &FA_x := FA_0 \quad FB_x := FB_0 \quad \text{Aproximaciones iniciales} \\
 &\text{Given} \\
 &\left(FC(FA_x, FB_x)^2 \cdot FD(FA_x, FB_x)^2 - KP_1 \cdot FA_x^3 \cdot FB_x \right) \cdot 10^0 = 0 \\
 &\left(FE(FA_x, FB_x) \cdot FF(FA_x, FB_x)^2 \cdot FT(FA_x, FB_x)^2 - KP_2 \cdot FA_x^2 \cdot FC(FA_x, FB_x) \cdot FD(FA_x, FB_x)^2 \cdot PT^2 \right) \cdot 10^0 = 0 \\
 &FA_x > 0 \quad FA_x \leq FA_0 \quad FB_x > 0 \quad FB_x \leq FB_0 \\
 &\begin{pmatrix} FA_{eq} \\ FB_{eq} \end{pmatrix} := \text{Find}(FA_x, FB_x) \quad \begin{pmatrix} FA_{eq} \\ FB_{eq} \end{pmatrix} = \begin{pmatrix} 0.752 \\ 9.455 \end{pmatrix} \text{ Kmoles/min} \quad +
 \end{aligned}$$

Puede notarse que entre las instrucciones del *Given* y *Find*, pueden incluirse convenientemente restricciones como asegurar que los flujos buscados sean positivos y, para el caso de los reactivos, menores a lo alimentado. Una vez que se tienen los resultados se debe verificar tengan significado físico calculando el resto de los flujos y comprobando que todos sean positivos. Además, podemos comprobar si necesario modificar el exponente del factor 10^0 comprobando el orden de magnitud del primero de los términos de cada ecuación implícita:

$$\begin{aligned}
 &FC(FA_{eq}, FA_{eq})^2 \cdot FD(FA_{eq}, FA_{eq})^2 = 2.665 \times 10^6 \\
 &FE(FA_{eq}, FB_{eq}) \cdot FF(FA_{eq}, FB_{eq})^2 \cdot FT(FA_{eq}, FB_{eq})^2 = 871.144
 \end{aligned}$$

Como ambos son suficientemente grandes concluimos que, para este caso en particular, no es necesario cambiar el exponente en el factor $\times 10^0$, si el orden de magnitud de alguno de estos términos fuera muy pequeño, por ejemplo, 10^{-3} o 10^{-4} , entonces, sí sería conveniente aumentar ese exponente para controlar la tolerancia absoluta del criterio de convergencia.

Finalmente, calculamos el resto de los flujos y resultados deseados comprobando que la solución tenga significado físico (por ejemplo, todos los flujos dependientes calculados con los resultados también sean positivos). A continuación se incluyen sólo parte de los cálculos restantes como muestra.

$$FC_{eq} := FC(FA_{eq}, FB_{eq}) \quad FC_{eq} = 4.28395 \quad \text{Kmoles/min}$$

$$FT_{eq} := FT(FA_{eq}, FB_{eq}) \quad FT_{eq} = 20.3876$$

$$pA_{eq} := \frac{FA_{eq} \cdot PT}{FT_{eq}} \quad pA_{eq} = 0.0738 \quad \text{bar}$$

$$pC_{eq} := \frac{FC_{eq} \cdot PT}{FT_{eq}} \quad pC_{eq} = 0.4202$$

Seguimiento Homotópico

El seguimiento de la solución es necesario para sistemas de ecuaciones algebraicas no-lineales *rígidos*. También se puede adaptar para sistemas de ecuaciones diferenciales con condiciones en dos fronteras pues requieren de *Métodos de Disparo*. Una vez que se cuenta con una solución, ésta se emplea como aproximación inicial para un cambio pequeño de una o más de las variables (incógnitas del sistema de ecuaciones simultáneas) hasta llegar a la solución deseada. Tomamos el Ejemplo 2.7 y la ecuación G_2 de la página 76, definimos funciones y valores conocidos:

$$FA_0 := 10 \quad \text{Kmoles/min} \quad KP_1 := 55.108 \quad \text{adim} \quad PT := 2 \quad \text{bar}$$

$$FB_0 := 12 \quad KP_2 := 7.424 \quad \text{bar}^{\wedge}2 \quad \text{Nota: las unidades son simples letreros que se distinguen por una fuente diferente}$$

$$FC(FA, FB) := 37 + \frac{FA}{2} - \frac{7 \cdot FB}{2} \quad FF(FA, FB) := 3 \cdot FB - FA - 26$$

$$FD(FA, FB) := 50 + FA - 5 \cdot FB \quad FT(FA, FB) := 48 + FA - 3 \cdot FB$$

$$FE(FA, FB) := \frac{3}{2} \cdot FB - \frac{FA}{2} - 13$$

Como cálculo intermedio definimos la función implícita F_2 y la empleamos dentro del comando `root` para encontrar una raíz y asignar el resultado a la función G_2 . Es importante notar que FB_x es la aproximación inicial que se introduce directamente como argumento de la función G_2 y no como una situación típica donde simplemente se define antes y la función implícita sólo tiene un argumento, FA en este caso. La parte clave en este procedimiento es que la aproximación inicial entre como argumento para que posteriormente se pueda ir almacenando su valor para irlo retomando durante el seguimiento.

$$F_2(FA, FB) := \left(\frac{FE(FA, FB) \cdot FF(FA, FB)^2 \cdot FT(FA, FB)^2}{FA^2 \cdot FC(FA, FB) \cdot FD(FA, FB)^2 \cdot PT^2} - KP_2 \right) \cdot 10^0$$

$$G_2(FA, FB_x) := \text{root}(F_2(FA, FB_x), FB_x)$$

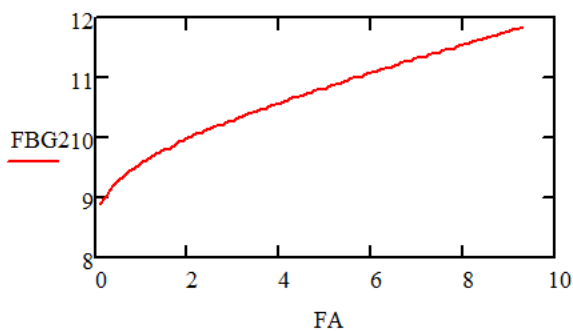
Ahora es necesario utilizar *Barra de Programación* de MathCad. Asignamos una primera aproximación inicial para FB_x , un valor de 0 al subíndice `ind` y hacemos el ciclo `for` para variar FA desde 0.1 hasta

FA0 con incrementos de 0.1 (0.2 - 0.1). No se comienza con 0 porque la función se vuelve indeterminada al estar FA en el denominador del primer término. Cada valor de FA se va guardando en columna 0 de la matriz M y para cada valor se resuelve para encontrar FB. El valor se guarda en la columna 1, pero antes se actualizar el valor de la aproximación inicial FBx. El ciclo termina incrementando el subíndice ind. El último M que aparece es para transferir resultados a la función que se desea definir, M también en este caso que corresponderá a una matriz. Para facilitar la interpretación la columna 0 de M se asigna a la variable FA que será un vector y la columna 1 a FBG2 para indicar FB calculado a partir de la función G2 del ejemplo.

```

FBx := 10
M := | FBx ← FBx
      | ind ← 0
      | for FA ∈ 0.1, 0.2 .. FA0
      | | Mind,0 ← FA
      | | FBx ← G2(FA, FBx)
      | | Mind,1 ← FBx
      | | ind ← ind + 1
      | M
FA := M<0>   FBG2 := M<1>
    
```

En el procedimiento anterior al menos un par de pasos se podrían eliminar, pero se incluyeron de esta manera buscando facilitar en su interpretación. Ahora podemos hacer un gráfico con los resultados. Es importante mostrar aún con el seguimiento homotópico, por no haber *modificado* la ecuación implícita se tuvieron dificultades numéricas para valores de FA mayores a 9.5 que convergen a números imaginarios y la indeterminación para FA igual a 0.



ind := 94 FA_{ind} = 9.5 FBG2_{ind} = 19.055 - 43.914i

G2(0, FBx) = ■

Divide by zero.

Integración

Este método es muy fácil de usar. Simplemente tenemos que utilizar el editor para escribir la fórmula con la Barra de Integración y el paquete la resuelve. MathCad lleva cierto control del error numérico de manera que no hay que definir el número de pasos de integración. A continuación resolvemos la integral del Ejemplo 4.3 que se encuentra en la página 188:

$$\begin{aligned}
 CA_0 &:= 0.15 \text{ M} & k &:= 0.0405 \frac{\text{mol}^{0.5}}{\text{lt}^{0.5} \text{ min}} & k_r &:= 0.0011 \frac{\text{lt}^{0.5}}{\text{mol}^{0.5} \text{ min}} \\
 r_{A\text{neg}}(f_A) &:= k \cdot \frac{CA_0 \cdot (1 - f_A)}{\sqrt{0.5 \cdot CA_0 \cdot f_A}} - k_r \cdot (1.5 \cdot CA_0 \cdot f_A)^{1.5} \\
 f_{A1} &:= 0.98 \\
 t_1 &:= \int_0^{f_{A1}} \frac{CA_0}{r_{A\text{neg}}(f_A)} df_A & t_1 &= 24.077 \text{ min}
 \end{aligned}$$

No es posible definir una función o variable que incluya el signo negativo, $(-r_A)$, entonces para dejar en claro que es el *negativo* de la velocidad de A se optó por definirla como $r_{A\text{neg}}(f_A)$. Por otro lado, las unidades hemos recomendado dejarlas indicadas con simples letreros siempre que sea posible. En este caso hemos optado por emplear el editor de ecuaciones de MathCad para las unidades de las constantes de velocidad, pero con un *click derecho* sobre la ecuación se seleccionó “Deshabilitar la Evaluación”. El pequeño rectángulo en negro que aparecen con ambas unidades indica que están deshabilitados y se convierten en letreros. Si no se deshabilitaran, la hoja de cálculo pudiera volverse más lenta o marcar algún posible error porque se estarían dejando inconclusas ecuaciones o definiciones.

Runge-Kutta

Este método es el *caballito de batalla* para el diseño de reactores. Con este método también podemos realizar la integración numérica y, como discutimos al final de la sección “Implicaciones de la Funcionalidad de la ecuación de diseño” en la página 180, podemos evitar métodos iterativos al optar por este método en lugar de alguno de integración más convencional. Este método integra ecuaciones diferenciales ordinarias simultáneas de primer orden. Con ligeras modificaciones y aumento del número de ecuaciones podemos resolver diferenciales de órdenes superiores. Cuando las expresiones cinéticas son de moderada dificultad debemos utilizar este método para resolver problemas de reactores por lotes y reactores de flujo pistón.

Vamos a retomar el Ejemplo 4.8 y resolver para un tiempo de espacial de 100 min con las tres ecuaciones de diseño presentadas en la página 210. Como son tres reacciones independientes, tres es el número mínimo de ecuaciones de diseño que se debe resolver simultáneamente (salvo el caso especial en que una de esas ecuaciones sólo depende de sí misma).

Comenzamos la hoja de cálculo definiendo algunos valores y funciones conocidas:

$$\begin{aligned}
 EA_1 &:= 125000 \text{ J/mol} & A_1 &:= 5.5 \cdot 10^{19} \text{ lt/mol min} \\
 EA_2 &:= 158000 & A_2 &:= 7.2 \cdot 10^{24} \text{ min}^{-1} \\
 EA_3 &:= 98000 & A_3 &:= 9.1 \cdot 10^{14} \text{ lt/mol min} \\
 T_{\text{prueba}} &:= 30.0 \text{ }^\circ\text{C} & & \text{Tprueba se cambia durante la optimación manual} \\
 CA_0 &:= 0.35 \text{ M} & &
 \end{aligned}$$

Como se nos pide repetir el caso para distintas temperaturas, incluimos funciones para las constantes de velocidad de reacción y de equilibrio de la primer reacción:

$$k_1(T) := A_1 \cdot e^{\frac{-EA_1}{8.314 \cdot (T+273.15)}} \quad k_2(T) := A_2 \cdot e^{\frac{-EA_2}{8.314 \cdot (T+273.15)}}$$

$$k_3(T) := A_3 \cdot e^{\frac{-EA_3}{8.314 \cdot (T+273.15)}} \quad K(T) := 9.5 \cdot 10^{21} \cdot e^{\frac{-141000}{8.314 \cdot (T+273.15)}}$$

Cuando sea requerido, se incluyen las funciones para las variables estequiométricamente dependientes. Para este caso sólo se emplea CC y, aunque se pudiera escribir sólo como una función de CB, aquí se puso también como función de las concentraciones de los otros dos componentes independientes buscando generalizar el ejemplo. Las funciones de las velocidades de reacción también podrían tener a la temperatura como parte de los argumentos, como será indispensable para operación no-isotérmica, pero en este ejemplo se calculan directamente con el valor previamente definida, Tprueba, que permanece constante por las condiciones isotérmicas del reactor.

$$CC(CA, CB, CE) := 3 \cdot CB$$

$$r_1(CA, CB, CE) := k_1(Tprueba) \cdot \left(CA^2 - \frac{CB \cdot CC(CA, CB, CE)^3}{K(Tprueba)} \right)$$

$$r_2(CA, CB, CE) := k_2(Tprueba) \cdot CA^2$$

$$r_3(CA, CB, CE) := k_3(Tprueba) \cdot CA \cdot CE$$

Una alternativa que evitaría estar evaluando k_1 , k_2 , k_3 y K cada vez que se evalúa una velocidad de reacción con las funciones anteriores es simplemente calcularlos previamente, por ejemplo, $k1const := k_1(Tprueba)$, y emplear dichas variables directamente dentro de funciones de velocidad de reacción.

Ahora viene lo interesante y fino, debemos incluir un vector con las *condiciones iniciales*, y la parte donde realmente se introduce el *modelo matemático* que es un vector de funciones. Para este reactor tubular en fase líquida les hemos llamado C0 y D(τ, C), respectivamente. Ambos elementos deben corresponder, es decir, en orden estricto, cada renglón de C0 representa la condición inicial de la ecuación de diseño correspondiente en D(τ, C). Debemos recordar que MathCad numera sus renglones y columnas comenzando con 0. Para este caso, los renglones 0, 1 y 2 corresponden a A, B y E, que son los componentes independientes.

$$C_0 := \begin{pmatrix} CA_0 \\ 0 \\ 0 \end{pmatrix} \quad D(\tau, C) := \begin{pmatrix} -2 \cdot r_1(C_0, C_1, C_2) - r_2(C_0, C_1, C_2) - r_3(C_0, C_1, C_2) \\ r_1(C_0, C_1, C_2) \\ r_2(C_0, C_1, C_2) - r_3(C_0, C_1, C_2) \end{pmatrix}$$

C0 es simplemente el nombre de la variable. Al definir las ecuaciones de diseño dentro del lado derecho de D(τ, C) notamos que aparecen elementos del vector C con subíndices 0, 1 y 2. No podemos simplemente escribir CA, CB y CE y esperar que MathCad lo interprete correctamente. C es el vector de las variables estequiométricamente independientes, C_A , C_B y C_E pero a las que necesariamente debemos referirnos como C_0 , C_1 y C_2 dentro de las ecuaciones de diseño, respectivamente. Los subíndices son “reales” y deben obtenerse de la Barra de Matrices y no de “apariencia” resultado de teclear “.” para generar el subíndice. No debemos confundir C_0 , ce-subíndice-real-cero, con C0, ce-cero, y menos con C0, ce-o-mayúscula.

Los perfiles los obtenemos empleando el método de Runge-Kutta con la función `rkfixed` con los argumentos: C_0 es el vector con las condiciones iniciales, 0 es el límite inferior de la variable independiente τ , τ_1 es el límite superior hasta donde se integran las ecuaciones diferenciales, `nPasos` es el número de pasos de integración y D la variable donde se definen las ecuaciones diferencias pero sin los argumentos (no escribimos $D(\tau, C)$ sino simplemente D).

```
nPasos := 200      τ1 := 200 min
Perfil := rkfixed(C0, 0, τ1, nPasos, D)
```

El resultado es una matriz y se asignó a la variable `Perfil`. Los renglones corresponden a resultados intermedios de cada paso de integración. La columna 0 corresponde a la variable independiente, τ . Las columnas subsecuentes son las variables dependientes de cada ecuación diferencial definidas en $D(\tau, C)$. No debe confundirnos que aunque son dependientes de τ también son estequiométricamente independientes, esto es, independientes para fines puramente estequiométricos. Sin embargo, los subíndices referentes a las columnas dentro de `Perfil` están recorridos un lugar respecto al vector C , por ejemplo, C_A está asociado a C_0 mientras que su perfil está en la columna 1.

	0	1	2	3
Perfil =	197	0.193	0.051	0.033
	198	0.193	0.051	0.033
	199	0.193	0.051	0.034
	200	0.193	0.051	...

$\tau_p := \text{Perfil}^{(0)}$ $CA_p := \text{Perfil}^{(1)}$ $CB_p := \text{Perfil}^{(2)}$ $CE_p := \text{Perfil}^{(3)}$

Para extraer elementos de este matriz empleamos subíndices: primero el renglón seguido de un coma y luego número de la columna. Por ejemplo, para valor C_A al final escribiríamos $CA1 := \text{Perfil}_{nPasos, 1}$. Sin embargo, se optó por otra opción que hará más sencilla la interpretación de los gráficos. La instrucción “<>” sacada de la Barra de Matrices permite asignar columnas específicas de variables (matrices) lo que resultará siendo un vector. Se ha incluido una “p” en los nombres para recordar que se trata de perfiles.

A continuación se extraen algunos resultados y se realizan algunas cálculos. Notamos la conveniencia de haber empleado `nPasos` dentro de los argumentos de `rkfixed` en lugar de simplemente haber escrito 200. De esta manera, si requerimos cambiar el número de pasos solamente lo haríamos en un solo lugar, a diferencia de tener que cambiar ese 200 de manera individual en varios lados además del riesgo de olvidar cambiar alguno.

```
CA1 := CAP_nPasos   CB1 := CBp_nPasos   CE1 := CEP_nPasos
CC1 := CC(CA1, CB1, CE1)
```

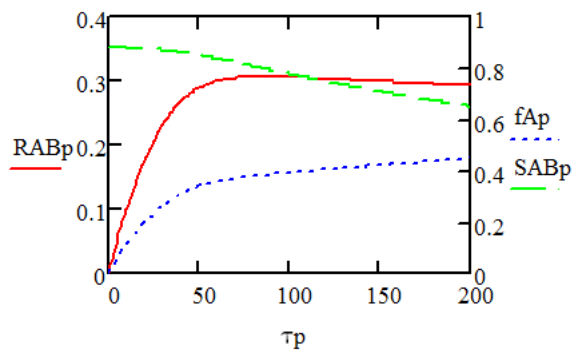
$CA1 = 0.1926$	M	$fA1 := \frac{CA0 - CA1}{CA0}$	$fA1 = 0.4498$
$CB1 = 0.0512$	M	$SAB1 := \frac{2CB1}{CA0 - CA1}$	$SAB1 = 0.65$
$CE1 = 0.0337$	M	$RAB1 := SAB1 \cdot fA1$	$RAB1 = 0.293$
$CC1 = 0.1536$	M		

MathCad permite hacer ciertas operaciones con variables asociadas a vectores. Por ejemplo, sumar o restarles una constante o multiplicarlas o dividir las por una constante. Estas operaciones las empleamos a continuación para obtener los perfiles fAp y $RABp$. Dependiendo de la versión de MathCad, algunas

sí permiten multiplicación o división “sencilla” directa entre elementos correspondientes a un vector con los de otro vector. No estamos hablando operaciones matriciales producto-punto ni producto-cruz. Por evitar confusiones o diferencias de compatibilidad de las operaciones es mejor realizar las operaciones para SABp a través de la Barra de Programación. Notamos que habría una indeterminación a $\tau = 0$ para la selectividad, $\frac{2 \times 0}{C_{A0} - C_{A0}}$. En lugar de calcular el límite, simplemente se le asignó el resultado del siguiente paso de integración que debe corresponder a dicho límite.

$$f_{Ap} := \frac{CA_0 - CA_p}{CA_0} \quad SAB_p := \left\{ \begin{array}{l} \text{for ind} \in 1 \dots nPasos \\ SAB_{p_ind} \leftarrow \frac{2 \cdot CB_{p_ind}}{CA_0 - CA_{p_ind}} \\ SAB_{p_0} \leftarrow SAB_{p_1} \\ SAB_p \end{array} \right. \quad RAB_p := \frac{2 \cdot CB_p}{CA_0}$$

Por último, de una manera sencilla podemos mostrar los perfiles en gráficos, por ejemplo:



Runge-Kutta para Una Sola Ecuación Diferencial

Aunque existen otros métodos dentro de *MathCad*, algunos para una sola ecuación diferencial. Sugerimos emplear lo visto en esta sección definiendo C0 y D(τ , C) como si se tratara de dos ecuaciones diferenciales, pero escribiendo 0 en el segundo renglón de ambos. Simplemente nos concentramos en los resultados que nos interesan e ignoramos aquellos de la segunda ecuación diferencial que serán todos 0.

Ecuaciones Diferenciales Ordinarias con Condiciones en Dos Fronteras

Este tipo de ecuaciones aparecen principalmente en este texto al considerar los efectos internos. Los métodos de solución son conocidos como *métodos de disparo* porque involucran suponer ciertas condiciones en una frontera, integrar con un método similar al Runge-Kutta para calcular los valores correspondientes a la otra frontera, comparar e iterar hasta alcanzar convergencia. En esta sección se trata un problema en que se debe converger una sola variable. El Apéndice ?? presenta este mismo procedimiento pero para la convergencia de varias variables en una de las fronteras.

La solución de estos problemas con *MathCad* debe realizarse en dos etapas: 1) Buscar con la función sbval todas las condiciones en una frontera que satisfagan a las restantes de la otra frontera; y 2) Utilizar el Runge-Kutta para integrar las ecuaciones con los resultados de la función sbval.

Vamos a resolver las ecuaciones diferenciales que se emplearon para generar la curva del factor de efectividad para una esfera y $n = 2$ de la Figura 9.3. En particular se va a evaluar el factor de

efectividad para un módulo de Thiele igual a 2. La solución matemática involucra a la Ecuación 9.16 con el módulo de Thiele definido con la Ecuación 9.15. Las condiciones en dos fronteras son: $\hat{C} = 1$ en $\hat{r} = 1$ y $d\hat{C}/d\hat{r} = 0$ en $\hat{r} = 0$. La ecuación diferencial de segundo orden se transforma en dos ecuaciones ordinarias de primer orden:

$$\frac{d\hat{C}}{d\hat{r}} = Y$$

$$\frac{dY}{d\hat{r}} = \begin{cases} -\frac{1}{3}\Phi^2 \hat{C}^2 & \text{si } \hat{r} = 0 \\ -\frac{2}{r}Y - \Phi^2 \hat{C}^2 & \text{si } \hat{r} > 0 \end{cases}$$

Comencemos declarando algunas constantes como el módulo de Thiele, el orden de reacción y los límites adimensionales de integración:

$\Phi := 2$ $n := 2$ $r0 := 0$ $r1 := 1$ **Radios adimensionales**

Ahora definimos un vector de las derivadas de \hat{C} y Y :

$$D(r,C) := \text{if } \left[r > 0.0001, \begin{bmatrix} \frac{-2}{r} \cdot C_0 + \Phi^2 \cdot (C_1)^n \\ C_0 \end{bmatrix}, \begin{bmatrix} \frac{\Phi^2 \cdot (C_1)^n}{3} \\ C_0 \end{bmatrix} \right] \quad \begin{array}{l} \text{Primero las ecuaciones de} \\ \text{los conocidos} \\ \text{en } r=0 \text{ y luego las que se} \\ \text{"disparan"} \end{array}$$

El orden es muy importante para la primera etapa. Primero deben incluirse las ecuaciones diferenciales para las que se conocen las condiciones en la primera frontera. Notemos que el orden del vector $D(r,C)$ está alterado respecto a las ecuaciones diferenciales de las que partimos. La codificación anterior tiene el condicional `if` para evitar una singularidad en $\hat{r} = 0$. Debemos percatarnos además que en las expresiones anteriores que C_0 es Y y C_1 es \hat{C} . Para la primera etapa de solución no se incluye una tercera ecuación diferencial que calcule η porque en esta etapa solamente nos interesa obtener las condiciones en una frontera.

La función `sbval` requiere del vector $D(r,C)$ con primeras derivadas, el vector v que contiene las aproximaciones iniciales en la primer frontera de las condiciones especificadas en la segunda frontera. Como en nuestro ejemplo solamente se tiene una variable especificada en la segunda frontera por eso se utiliza un vector con una sola entrada (v_0). Se debe definir un vector `load(r1,v)` que incluya todas las condiciones en la primer frontera, incluyendo las supuestas. Es muy importante que primero se incluyan todas las conocidas y al último el vector v que corresponde a las supuestas. Para lograr este orden puede ser necesaria una reordenación para que correspondan los renglones del vector $D(r,C)$, como lo hemos hecho para este caso. Finalmente, el vector `score(r2,C)` (también de un solo elemento para este ejemplo) calcula los errores de las variables especificadas en la segunda frontera. La codificación en MathCad es:

$v_0 := 0.1$ Aproximación inicial variable de disparo (vector de un elemento)

$\text{load}(r0,v) := \begin{pmatrix} 0 \\ v_0 \end{pmatrix}$ Primero se deben incluir las conocidas y luego los desconocidos.

$\text{score}(r1,C) := C_1 - 1$ Criterio de convergencia, frontera en $r=1$

$S := \text{sbval}(v,r0,r1,D,\text{load},\text{score})$ $S_0 = 0.639$ Concentración adimensional en el centro

En las entradas anteriores se está suponiendo que $\hat{C} = C_1 = 0.1$ en $\hat{r} = 0$. El valor superior en $\text{load}(r1, v)$ porque $Y = C_0 = 0$ en $\hat{r} = 0$. Como la segunda especificación es $\hat{C} = C_1 = 1$ en $\hat{r} = 1$, el vector $\text{score}(r2, C)$ (de un elemento en este caso) por ello incluye la entrada $C_1 - 1$ de manera que sbva1 minimice dichos errores. El resultado de sbva1 es un vector con los valores en la primer frontera que equivalen o cumplen con los de la segunda. Por lo tanto, en este ejemplo al integrar las ecuaciones diferenciales con $\hat{C} = 0.639$ y $Y = 0$ en $\hat{r} = 0$ se obtiene $\hat{C} = 1$ en $\hat{r} = 1$.

Como segunda etapa de solución debemos realizar la integración con los resultados de la función sbva1 . Ahora sí vamos a incluir la ecuación que vaya calculando el factor de efectividad:

$$\frac{d\eta}{d\hat{r}} = 3\hat{r}^2 \hat{C}^2$$

Rescribimos el vector $D(r, C)$. El orden corresponde al de las ecuaciones diferenciales del inicio de esta sección y, por las razones discutidas en la Sección 9.3, se incluye como tercera la ecuación anterior:

$$D(r, C) := \text{if } r > 0.0001, \begin{bmatrix} C_1 \\ \frac{-2}{r} \cdot C_1 + \Phi^2 \cdot (C_0)^n \\ 3 \cdot r^2 \cdot (C_0)^n \end{bmatrix}, \begin{bmatrix} C_1 \\ \frac{1}{3} \cdot \Phi^2 \cdot (C_0)^n \\ 3 \cdot r^2 \cdot (C_0)^n \end{bmatrix}$$

Escribimos las condiciones iniciales y notemos que se han incluido los resultados de la primera etapa de solución. Para este ejemplo se seleccionó un método de Runge-Kutta adaptivo, Rkadapt , que ajusta el tamaño de los pasos de integración:

$$C0 := \begin{pmatrix} S_0 \\ 0 \\ 0 \end{pmatrix} \quad \begin{array}{l} \text{nPasos} := 1000 \\ \text{Perfil} := \text{Rkadapt}(C0, r0, r1, \text{nPasos}, D) \end{array}$$

$$\eta := \text{Perfil}_{\text{nPasos}, 3} \quad \boxed{\eta = 0.712}$$

$$Cs := \text{Perfil}_{\text{nPasos}, 1} \quad Cs = 1 \quad \dots \text{como comprobación}$$

El único resultado que nos interesa corresponde a los subíndices $(\text{nPasos}, 3)$ porque se emplearon 1000 pasos de integración y este valor como subíndice representa a la frontera en $\hat{r} = 1$. Valores del segundo subíndice de 0, 1, 2, y 3 se refieren \hat{r} (la variable independiente), \hat{C} , Y y η , respectivamente. Por lo tanto, el factor de efectividad para una esfera, $n = 2$ y $\Phi = 2$ es 0.712.

©Dr. Fernando Tiscareño Lechuga
Departamento de Ingeniería Química
Instituto Tecnológico de Celaya

Versión Preliminar para Segunda Edición del 28 de agosto de 2018

Las páginas y numeraciones de ecuaciones, capítulos, figuras y secciones no necesariamente corresponderán en esta versión preliminar.

