ASCEND: un simulador de procesos orientado a ecuaciones al alcance de los estudiantes de Ingeniería Química

C. Planelles, J.D. Medrano, R. Ruiz-Femenia y J. A. Caballero

Instituto Universitario de Ingeniería de Procesos Químicos. Universidad de Alicante. Carretera de San Vicente s/n. 03690 – San Vicente del Raspeig. Alicante.

Introducción

ASCEND ("*Advanced System for Computations in Engineering Design*") [1] es un programa de modelado matemático general que se puede utilizar como un simulador de procesos orientado a ecuaciones, ya que incluye una base de datos y modelos termodinámicos para la determinación de las propiedades de los compuestos [2]. Aparte de su carácter gratuito, la gran ventaja que presenta es su transparencia y flexibilidad, permitiendo al usuario, por ejemplo, comprobar la estructura del sistema de ecuaciones no lineales a través de su matriz de incidencia. Además, al igual que el resto de simuladores de procesos orientados a ecuaciones, supera ampliamente a los simuladores modulares en su capacidad para optimizar el diagrama de flujo. ASCEND puede conectarse con los "*solvers*" de optimización más avanzados para problemas no lineales, como CONOPT o IPOPT. Dentro de la asignatura "Simulación, Diseño y Optimización de Procesos Químicos" del grado en Ingeniería Química de la Universidad de Alicante, se estudian los algoritmos (por ejemplo, SPK1 o RANKI) para resolver de forma eficaz los sistemas de ecuaciones dispersos característicos de los diagramas de flujo de procesos químicos. Siguiendo iniciativas como la de la Universidad de Alabama, en la UA hemos empezado a trabajar con este simulador, con el objetivo de utilizar ASCEND para la docencia de los simuladores orientados a ecuaciones, y así posibilitar que, el alumno reciba otro enfoque complementario al de "caja negra", característico de los simuladores modulares, que le permita visualizar el sistema de ecuaciones que modelan el proceso y comprender como afecta su estructura a la resolución del problema.

Ejemplo 1 Sistema sencillo de ecuaciones

```
REQUIRE "system.a4l";
MODEL my_model;
   (* Variables *)
   x, y, z, d, a IS_A solver_var;
   (* Equations *)
   EQ1: x^5 + y*x + = a*d;
   EQ2: x*z + 2*z = 5;
   EQ3: 5*x + 2*y - 3*z*z = -0.3;
METHODS
   METHOD specify;
      FIX d;
   END specify;
   METHOD values;
      d := 1;
   END values;
   METHOD on_load;
      RUN specify;
      RUN values;
   END on_load;
END my_model;
```

- Comienza la descripción del modelo, en este caso, de nombre "my_model".
- Definición de todas las variables que aparecen en el sistema.
- Los métodos son órdenes que se deben ejecutar antes de la resolución del problema.
- El método "values", asigna a *d* un valor de 1, complementando a "specify" para convertir d en una constante.
- Con "on_load", los métodos seleccionados cargan a la vez que el modelo.
- Llamada a la librería "system.a4l", donde se almacena la información que requiere ASCEND para crear variables.
- Para añadir un comentario, se escribe entre (* *).
- Definición de las ecuaciones que componen el modelo.
- El método "specify", fija las variables *d* y *a*. Se podría especificar *d = 1;* antes del apartado METHODS, sin embargo, *d* sería tratada como una variable y la expresión como una ecuación adicional.

Para la resolución de este modelo de 3 ecuaciones, existen inicialmente 5 variables. La variable *d* queda definida como una constante utilizando los métodos "**specify**" y "**values**". Por otro lado, se fijará el valor de *a* y se utilizarán las herramientas "**Observe**" y "**Study**" para fijar el valor de la variable *a* y resolver el sistema sucesivas veces variando su valor. La figura 1 muestra como evolucionan los valores (*x*, *y*, *z*) solución del sistema en función de la variable *a*.

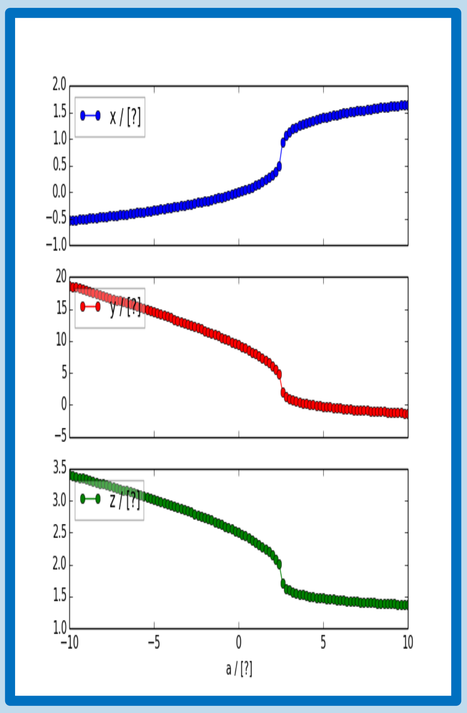

Figura 1. Resultados del sistema de (hacia abajo) *x*, *y*, *z* variando el valor de *a* de -10 a 10.

Además de estas funciones, ASCEND posee la capacidad de mostrar el dígrafo del sistema (figura 2) así como de representar la matriz de incidencia (figura 3) asociada al mismo.

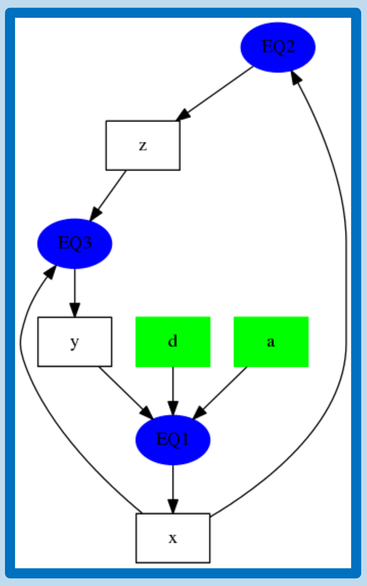

Figura 2. Dígrafo del sistema.

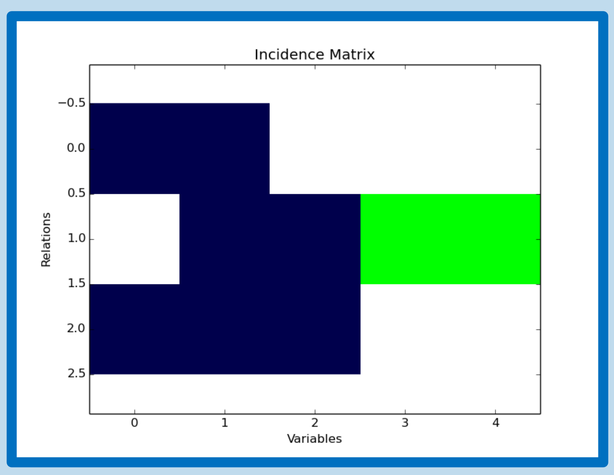

Figura 3. Matriz de incidencia.

Ejemplo 2 Propiedades termodinámicas en una mezcla binaria en equilibrio

```
REQUIRE "thermodynamics.a4l";
REQUIRE "stream_holdup.a4l";
MODEL binary_LV
REFINES cmumodel;
   cd IS_A components_data
(['water','ethanol'],'water');
   pd IS_A phases_data('VL',
'ideal_vapor_mixture',
'UNIFAC_liquid_mixture', 'none');
   equilibrated IS_A boolean;
   phases ALIASES pd.phases;
   FOR j IN phases CREATE
      smt[j] IS_A
      select_mixture_type(cd,
      pd.phase_type[j]);
   END FOR;
   FOR j IN phases CREATE
      phase[j] ALIASES smt[j].phase;
   END FOR;
   state IS_A thermodynamics(cd,
   pd,phase, equilibrated);
METHODS
(...)
   METHOD values;
      state.P := 1 {atm};
      state.T := 300 {K};
      state.y['ethanol'] := 0.5;
      equilibrated := TRUE;
   END values;
   METHOD reset_Px;
      equilibrated := TRUE;
      RUN state.specify;
      state.T.fixed := FALSE
   state.phase['liquid1'].y['water'].fi
xed := TRUE;
      state.P := 1 {atm};
      state.y['ethanol'] := 0.5;
      state.phase['liquid1'].y['water']
:= 0.6;
   END reset_Px;
END binay_LV;
```

- Librería que contiene modelos de estado estacionario para corrientes de V, L y LV.
- El modelo "binary_LV", hereda todo el código del modelo padre "cmumodel", que en este caso únicamente proporciona un directorio raíz.
- Información sobre el sistema de estudio (M, L, V, VL, LL, VLL) y los modelos termodinámicos a utilizar en cada fase.
- Bucle para la creación del vector *smt* con número de elementos igual al número de fases *j* de *phases*, y que contiene toda la información referente a las mismas.
- Cálculo de las propiedades termodinámicas y resolución del sistema para los valores de presión y temperatura, fracción molar establecidas en el método correspondiente.
- Método que sobreescribe a "values" cuando se ejecuta, liberando la variable temperatura y fijando la composición en el líquido. El modelo calculará la temperatura, la fracción de vapor y su composición.
- Esta librería contiene la información sobre los componentes y sus propiedades, ecuaciones de estado y modelos termodinámicos, además de todo lo contenido en "system.a4l".
- Constante que contiene diversas propiedades del agua y el etanol tales como energías de formación estándar o propiedades críticas.
- Variable booleana que establece si el sistema se encuentra o no en equilibrio.
- Vector cuyo número de elementos es igual al número de fases especificadas en *pd*.
- Bucle para la creación del vector *smt* con número de elementos igual al número de fases *j* de *phases*, y que contiene toda la información referente a las mismas.
- Valores de presión, temperatura y composición global. *T* y *P* están definidas como temperatura y presión, por lo que se deben introducir las unidades. El modelo calculará fracción de vapor y líquido así como sus composiciones.

La matriz de incidencia para el ejemplo 2 se muestra en la figura 4. Con este ejemplo se puede demostrar el potencial de ASCEND, que es capaz no sólo de resolver de forma casi instantánea el sistema de 60 ecuaciones con 60 incógnitas, si no que además muestra el procedimiento que ha seguido para hacerlo.

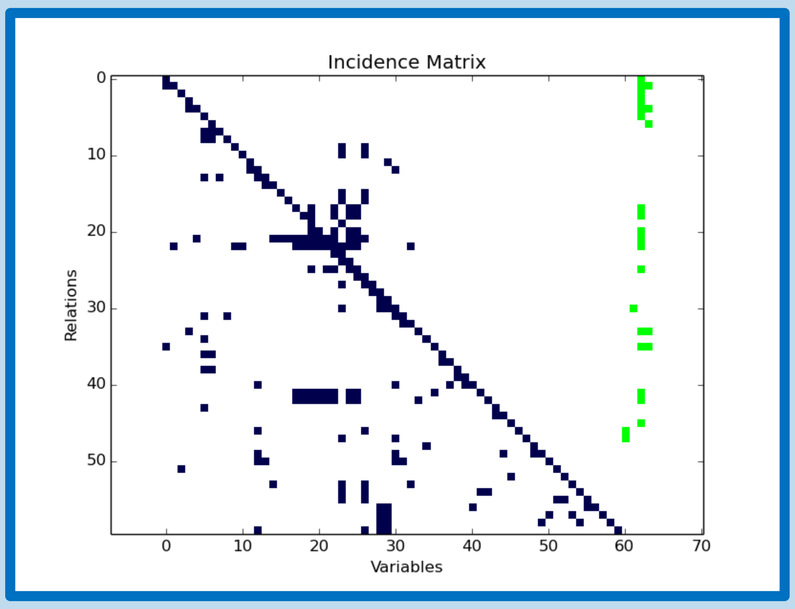

Figura 4. Matriz de incidencia.

Referencias

[1] Piela, P.; McKelvey, R.; Westerberg, A., (1992) "An introduction to ASCEND: its language and interactive environment," in *System Sciences, 1992. Proceedings of the Twenty-Fifth Hawaii International Conference on* System Sciences, vol.iii, no., pp.449-461 vol.3, 7-10.

[2] http://ascend4.org/Main_Page

Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Ciencia e Innovación del gobierno de España (CTQ2012-37039-C02-02).