

Evolución y Desafíos del Modelado y Simulación Orientados a Objetos

Ramón Pérez Vara

Director de Tesis

Prof. Sebastián Dormido Bencomo

Departamento de Informática y Automática
Universidad Nacional de Educación a Distancia (UNED)

Madrid, Mayo 2024



1

ÍNDICE

- 1. Modelado y Simulación Orientados a Objetos**
- 2. Particularidades de EcosimPro**
- 3. El Problema de la Simulación Estacionaria:**
 - La Rasgadura (tearing) de Lazos Algebraicos
- 4. El Problema de la Simulación Transitoria:**
 - Utilización de Formulaciones Diferentes (no DAE's)
 - Transitorios Hidráulicos por el Método de las Características (MOC)

2

1. MODELADO Y SIMULACIÓN ORIENTADO A OBJETOS

3

Modelado y Simulación Orientados a Objetos: Contexto y Desafíos 4/35

Tipos	Ejemplos	Ventajas	Desventajas
Herramientas Específicas	Spice Impulse Aspen Thermoflex	<ul style="list-style-type: none"> Facilidad uso Eficiencia Precisión Robustez Transitorios y Estacionarios 	<ul style="list-style-type: none"> Desarrollo Costoso Cajas negras Limitadas a un dominio Falta de flexibilidad para su extensión
Entornos de Simulación Orientados a Objeto	Modelica EcosimPro Amesim	<ul style="list-style-type: none"> Desarrollo Económico Generalidad Multidisciplinarios Flexibilidad Transitorias 	<ul style="list-style-type: none"> Dificultad de uso Ineficientes (obligan al uso de DAE's) Inadecuadas para estacionarios

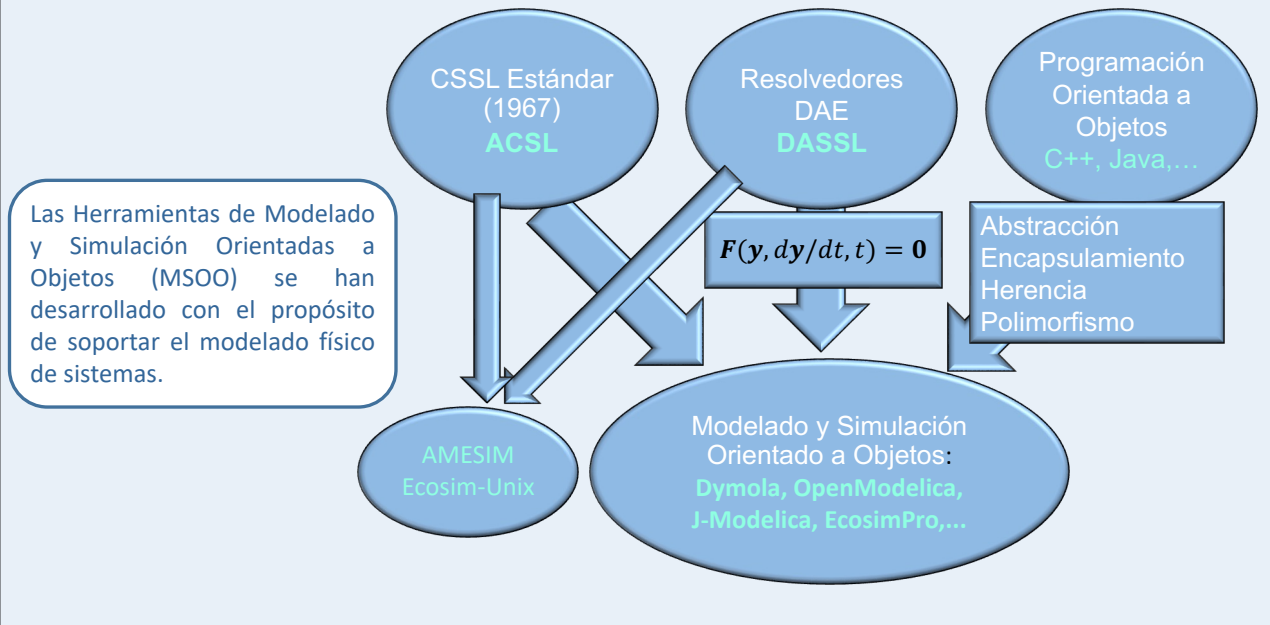
DESAFÍO 1

¿Es viable desarrollar aplicaciones de MSOO para campos específicos con la misma eficacia y facilidad de uso que programas de simulación de esos campos?

DESAFÍO 2

¿Se pueden implementar métodos de solución diferentes de los DAE's utilizando el lenguaje de simulación?

4



5

CARACTERÍSTICAS DE LAS HERRAMIENTAS DE MSOO

Lenguaje de tipo Declarativo para definir:

Declaración de Tipos de Puerto

```

PORT hydraulic
  SUM REAL Q UNITS "m3/s" "Volume flow"
  EQUAL REAL H UNITS "m" "Piezometric head"
  EQUAL REAL z UNITS "m" "Elevation"
  REAL P UNITS "MPa (g)" "Gauge Pressure"
CONTINUOUS
  P = rho * GRAV * (H-z) / 1.e6
END PORT
    
```

Declaración de Tipos de Componente

```

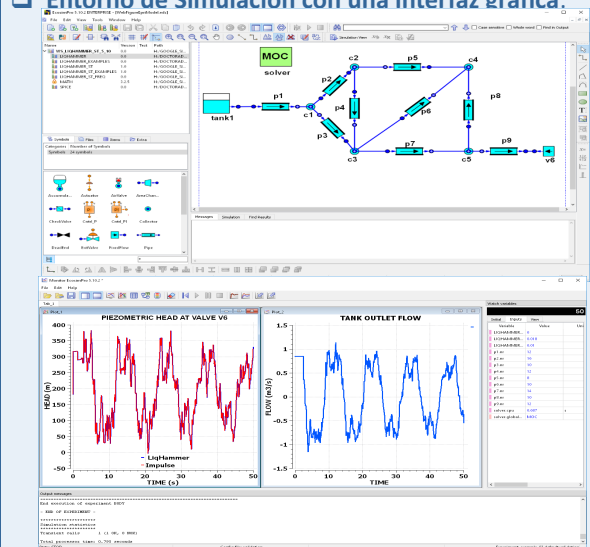
COMPONENT ExitValve
  PORTS
  IN hydraulic h_in
  DATA
    REAL Cd_A = 0.1 UNITS "m2" "Effective exit area"
    REAL z = 0. UNITS "m" "Valve elevation"
  DECLS
    BOUND REAL pos UNITS "p.u." "Valve position"
    REAL Q UNITS "m3/s" "Outlet flow"
  CONTINUOUS
    f_in.Q = Cd_A * pos * sqrt(2.*g) *sqrt(f_in.H- f_out.H)
    f_out.Q = Q
END COMPONENT
    
```

Ecuaciones algebraico diferenciales:

$$F(y, dy/dt, t) = 0$$

Discontinuidades: Sentencias When, Ecuaciones Condicionales y Asignaciones Retrasadas.

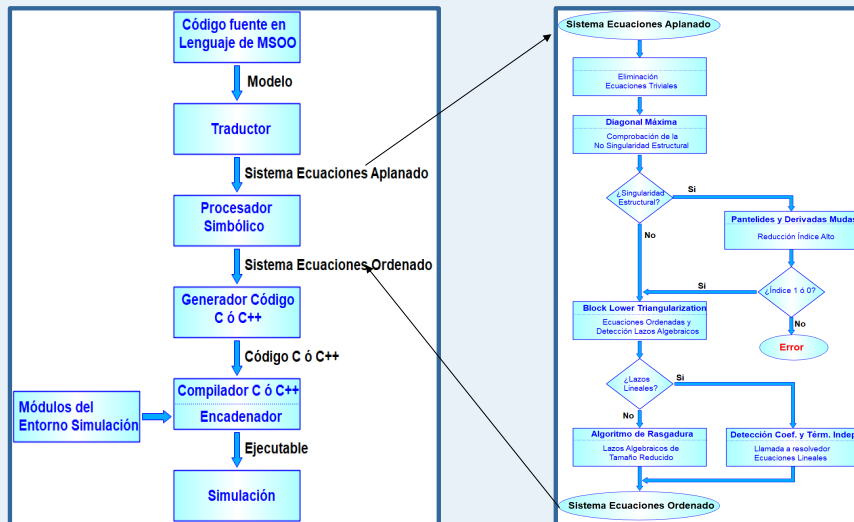
Entorno de Simulación con una interfaz gráfica:



6

CARACTERÍSTICAS DE LAS HERRAMIENTAS DE MSOO

Procesamiento Simbólico:



7

PROBLEMAS DEL MSOO

- ❑ Encapsulamiento de la información limitado a la fase de modelado.
- ❑ Diseño enfocado en el creador de librerías en lugar del usuario final: El lenguaje permite construir librerías de forma muy elegante.
- ❑ Inadecuado para modelos de gran tamaño: Procesamiento simbólico, número de líneas del código generado.
- ❑ Limitación en los métodos de solución a resolvedores de DAEs (Ecuaciones Algebraico Diferenciales) y ODEs (Ecuaciones Diferenciales Ordinarias).
- ❑ Problemas de integración transitoria: No linealidad y comportamientos tipo Zeno (chattering)
- ❑ Problemas de convergencia de las ecuaciones algebraicas: Debidos a la rasgadura

8

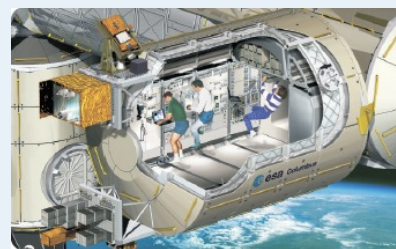
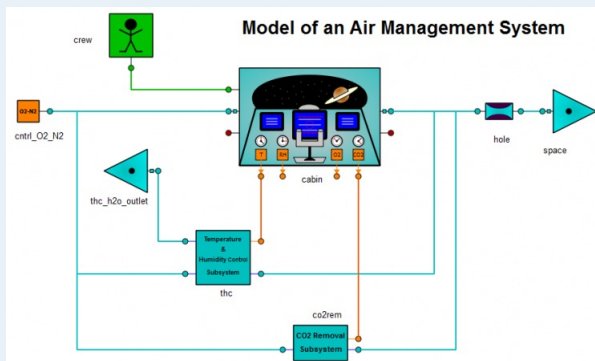
2. CARACTERÍSTICAS DE ECOSIMPRO

9

Características de EcosimPro: Visión General

10/35

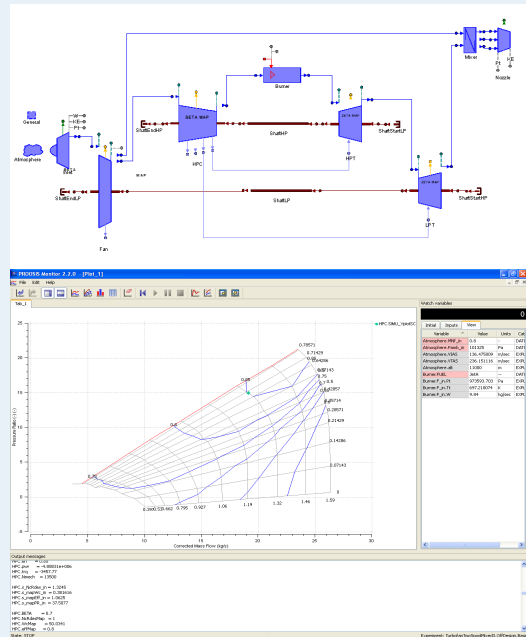
- ❑ EcosimPro es una herramienta de simulación genérica desarrollada por Empresarios Agrupados para la simulación dinámica de sistemas
- ❑ El cliente de la 1ª versión de EcosimPro fue la Agencia Espacial Europea.



- ❑ El propósito de la 1ª versión fue la simulación de los sistemas de control y soporte de vida de naves espaciales.
- ❑ EcosimPro evolucionó en una herramienta de simulación genérica aplicable a diversos tipos de sistemas.

10

- ❑ PROOSIS (PROpulsion Object-Oriented Simulation Software) es una herramienta de simulación derivada de EcosimPro para el modelado de aerorreactores y turbinas de gas aeronáuticas.
- ❑ PROOSIS nació en el marco de un proyecto europeo (VIVACE)
- ❑ PROOSIS añade a EcosimPro algunas capacidades necesarias para la simulación de aerorreactores:
- ❑ Biblioteca TURBO
 - Asistente de cálculo de diseño
 - Grupos de desigualdades
 - Manejo de mapas de componentes
 - Asistente de cálculo de sensibilidad



Características Principales EcosimPro

- ❑ Lenguaje de Modelado Orientado a Objeto y Estructurado en Bloques.
 - La entidad de modelado se designa **COMPONENT**
 - Un **COMPONENT** puede incluir los siguientes bloques: **PORTS**, **DATA**, **DECLS**, **TOPOLOGY**, **INIT**, **DISCRETE**, y **CONTINUOUS**
 - No proporciona Polimorfismo
 - Tres Sentencias Discretas: **WHEN**, **ZONE**, **Asignación Retrasada**.
- ❑ Código Secuencial (no ecuaciones) en el Bloque **INIT** de los componentes y dentro de las sentencias **WHEN**
- ❑ El lenguaje proporciona Funciones, Clases y Objetos (variables globales)
- ❑ Lenguaje de Definición de Experimentos

Bloques de los Componentes y Discontinuidades

```

COMPONENT Thermostat
PORTS
...
DATA
...
DECLS
...
TOPOLOGY
...
INIT
...
DISCRETE
...
CONTINUOUS
...
END COMPONENT
    
```

```

DISCRETE
WHEN(s_temp.signal < T_on) THEN
    tp_out.q = power
END WHEN
WHEN(s_temp.signal > T_off) THEN
    tp_out.q = 0
END WHEN
    
```

```

COMPONENT Sampler
DATA
    REAL t_sample = 0.5
DECLS
    BOOLEAN sample = TRUE
    DISCR REAL y = 0
DISCRETE
    WHEN (sample == TRUE) THEN
        sample = FALSE
        sample = TRUE AFTER t_sample
    END WHEN
END COMPONENT
    
```

13

Bloque INIT de ECOSIMPRO

- Ejecución secuencial al inicio de la simulación.
- Reasignación de valores entre bloques de distintos componentes y dentro del mismo.
- Fusión en un único bloque de código C++.
- El usuario puede definir PRIOTRIDADES
- En herencia, el bloque INIT de la clase derivada sigue al de la clase base.
- Mantenimiento de la estructura y ejecución de sentencias.
- Prioridades más altas se ejecutan primero.
- Función EXEC_INIT() para llamar la ejecución del bloque INIT, incluso desde dentro del propio bloque INIT

```

COMPONENT parent
DECLS
    DISCR REAL x = 0
    DISCR REAL y = 0
INIT
    y = y + 1
END COMPONENT
COMPONENT son IS_A parent
INIT
    x = y + 1
    y = x + 1
END COMPONENT
    
```

Bloque INIT ordenado del componente son

```

Y = Y + 1
x = y + 1
y = x + 1
    
```

Resultados: x = 2 y = 3

```

COMPONENT parent
DECLS
    DISCR REAL x = 0
    DISCR REAL y = 0
INIT PRIORITY 0
    y = y + 1
END COMPONENT
COMPONENT son IS_A parent
INIT PRIORITY 100
    x = y + 1
    y = x + 1
END COMPONENT
    
```

Bloque INIT ordenado del componente son

```

x = y + 1
y = x + 1
y = y + 1
    
```

Resultados: x = 1 y = 2

14

3. EL PROBLEMA DE LA SIMULACIÓN ESTACIONARIA

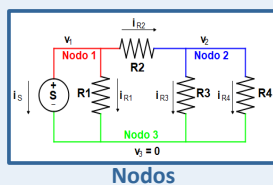
15

Simulación Estacionaria: Herramientas Específicas

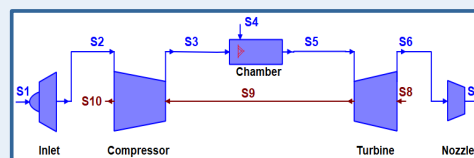
16/35

CARACTERÍSTICAS DE LA SIMULACIÓN ESTACIONARIA

- ❑ Basada en Sistemas de Ecuaciones Algebraicos.
- ❑ Conexionado Mediante “**Nodos**” o “**Corrientes**”.



Nodos



Corrientes

- ❑ Métodos de Solución “**Orientados a Ecuaciones**” o “**Modular Secuencial**”
- ❑ Rasgadura Sistemática del Sistema de Ecuaciones en las Herramientas Orientadas a Ecuaciones.

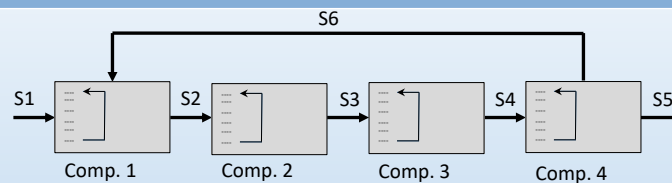
16

RASGADURA SISTEMÁTICA DEL SISTEMA DE ECUACIONES

- ❑ Selección Sistemática de Variables y Ecuaciones de Rasgadura en Herramientas Orientadas a Ecuaciones.
- ❑ No se busca minimizar el número de Ecuaciones Implícitas
- ❑ Requisitos de memoria y tiempo de CPU moderados (debido al uso de subrutinas de solución de sistemas de ecuaciones dispersos).
- ❑ Rasgadura en Programas basados en “Nodos”:
 - Variables de rasgadura: Voltajes (o variables esfuerzo) en los nodos.
 - Ecuaciones de rasgadura: Ecuaciones de suma de corrientes (o variables flujo) en los nodos igual a 0
- ❑ Rasgadura en Programas basados en “Streams”:
 - Variables de rasgadura: Variables independientes de las corrientes.
 - Ecuaciones de rasgadura: Ecuaciones de conservación y ciertas ecuaciones constitutivas en

17

MÉTODOS DE SOLUCIÓN : MODULAR SECUENCIAL



Ventajas

- Escalabilidad para Problemas Grandes:** Ideal para problemas de gran escala, al resolver los componentes secuencialmente.
- Modularidad del Código:** Facilita la inclusión de nuevos componentes.
- Facilidad en Programación y Mantenimiento:** Estructura modular que simplifica la programación y el mantenimiento del código.
- Control de Convergencia y Robustez:** Permite un control más simple de la convergencia y facilita la detección de errores.

Desventajas:

- Análisis Topológico Complejo:** Requiere un análisis topológico complejo para identificar reciclajes y seleccionar corrientes de rasgado.
- Lentitud en la Convergencia:** Puede tener velocidad de convergencia lenta, afectando la eficiencia de la simulación.
- Limitaciones en Diseño y Optimización:** No adecuada para problemas de diseño ni optimización.
- Inadecuada para modelos múltiples recirculaciones**
- Inadecuada para simulación transitoria**



18

RASGADURA DE LOS LAZOS ALGEBRAICOS

Problema

$$k_1 = k \left(\frac{T_0 + T_1}{2} \right) \quad q_1 = q_2 \quad k_2 = k \left(\frac{T_1 + T_2}{2} \right)$$

$$q_1 = k_1 \cdot (T_0 - T_1) \quad q_2 = k_2 \cdot (T_1 - T_2)$$

Condiciones Contorno: T_0 y T_2 conocidas, con $T_0 > T_2$

Rasgadura 1 (Sistema 1x1)

$$k_1 = k((T_0 + T_1)/2)$$

$$q_1 = k_1 \cdot (T_0 - T_1)$$

$$k_2 = k((T_1 + T_2)/2)$$

$$q_2 = k_2 \cdot (T_1 - T_2)$$

$$H_1(T_1) = q_1 - q_2 = 0$$

Correcta – Converge para cualquier T_1

Fuerza Bruta (Sistema 5x5)

$$F_1(T_1, q_1, q_2, k_1, k_2) = k_1 - k((T_0 + T_1)/2) = 0$$

$$F_2(T_1, q_1, q_2, k_1, k_2) = q_1 - k_1 \cdot (T_0 - T_1) = 0$$

$$F_3(T_1, q_1, q_2, k_1, k_2) = k_2 - k((T_1 + T_2)/2) = 0$$

$$F_4(T_1, q_1, q_2, k_1, k_2) = q_2 - k_2 \cdot (T_1 - T_2) = 0$$

$$F_5(T_1, q_1, q_2, k_1, k_2) = q_1 - q_2 = 0$$

Rasgadura 2 (Sistema 1x1)

$$k_1 = k((T_0 + T_1)/2)$$

$$q_1 = k_1 \cdot (T_0 - T_1)$$

$$q_2 = q_1$$

$$k_2 = k((T_1 + T_2)/2)$$

$$H_1(T_1) = k_2 - q_2/(T_1 - T_2) = 0$$

Errónea – Converge solo si $T_1 > T_2$

PROBLEMAS DE LA RASGADURA

❑ Rasgadura óptima es un problema NP completo.

- Las herramientas de MSOO implementan algoritmos heurísticos no públicos
- Limitación para la portabilidad de modelos entre herramientas de MSOO, incluso entre las que usan Modelica o entre diferentes versiones de la misma herramienta.

❑ Falta de robustez numérica:

- Dificultad de convergencia con mucha sensibilidad respecto los valores iniciales.
- Modos de Fallo (Baharev et al. 2017)
 - Sensibilidad extrema al rasgar.
 - Poca sensibilidad a cambios en la variable de rasgadura.
 - Fallos al despejar variables por operaciones indefinidas o múltiples soluciones.

❑ Variabilidad en la selección de variables y ecuaciones:

- Pequeños cambios en el modelo pueden causar cambios significativos en el conjunto de variables y ecuaciones de rasgadura, esto dificulta la inicialización de variables.

❑ Dificulta el uso de métodos numéricos dispersos:

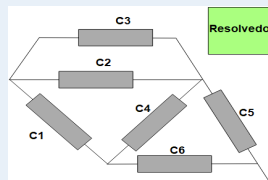
- Se pierde la estructura de la dispersión del sistema.

Nueva Aproximación para Estacionarios

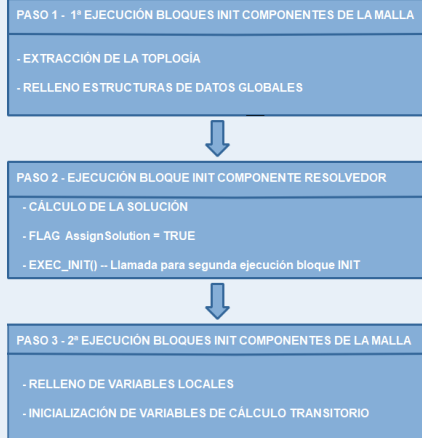
Pautas del Desarrollo

- **Entorno de Simulación Seleccionado:**
 - EcosimPro – Libertad del bloque INIT y capacidad para definir Clases
- **No interferir con las ecuaciones transitorias**
 - Implementación en bloque INIT
- **Flexibilidad en métodos de solución:**
 - Permitir el uso de métodos basados en ecuaciones o métodos de tipo modular secuencial.
- **Facilitar la resolución de sistemas de ecuaciones dispersos:**
 - Desarrollo de una librería de clases en EcosimPro para la solución de ecuaciones dispersas (interfaz con SUPERLU)

Modelo



Procedimiento



Extracción de la Topología con el bloque INIT

1. Uso de contadores para los componentes y asignación del nº de componente en el bloque INIT
2. Uso de un contador para los números de nodo y de variables enteras de puerto de tipo EQUAL para identificar los nodos
2. Asignación de los números de nodo en el bloque INIT de los componentes.

```

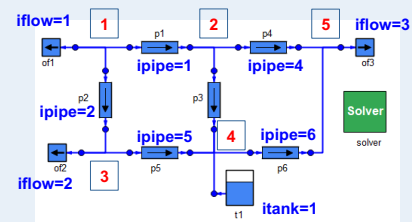
INTEGER npipe = 0 "Counter for the number of Pipes"
...
COMPONENT Pipe
DECLS
  INTEGER ipipe --pipe internal number
INIT
  IF (ipipe == 0) THEN
    npipe = npipe + 1
    ipipe = npipe
  END IF
  ...
END COMPONENT
    
```

```

INTEGER node = 0 "Counter for the number of hydraulic nodes"
PORT Hydraulic
EQUAL INTEGER inode = 0 --hydraulic nodes identifier
END PORT
    
```

```

COMPONENT Pipe
PORTS
  IN Hydraulic h_in "Hydraulic inlet port"
  OUT Hydraulic h_out "Hydraulic outlet port"
DECLS
  INTEGER ipipe = 0 --pipe identifier
INIT PRIORITY 100
  IF (ipipe == 0) THEN
    npipe = npipe + 1
    ipipe = npipe
    IF (h_in.inode == 0) THEN
      node = node + 1
      h_in.inode = node
    END IF
    IF (h_out.inode == 0) THEN
      node = node + 1
      h_out.inode = node
    END IF
  END IF
  ...
END COMPONENT
    
```



Pipe Name	ipipe	h_in node	h_out node
p1	1	1	2
p2	2	1	3
p3	3	2	4
p4	4	2	5
p5	5	3	4
p6	6	4	5
Tank Name	itank	h_out node	
t1	1		4
Outflow Name	iflow	h_in node	
of1	1	1	
of2	2	3	
of3	3	5	

Relleno Estructuras Datos Globales

1. Definición de una clase con los datos y atributos del Pipe

2. Declaración de un vector de objetos de la clase

3. Relleno de los atributos del objeto correspondiente

```

INTEGER npipe = 0 "Counter for the number of pipes"
CLASS PipeClass
DECLS
  INTEGER nd1 "Node at inlet"
  INTEGER nd2 "Node at outlet"
  REAL L UNITS "m" "Pipe length"
  REAL D UNITS "m" "Pipe inside diameter"
  REAL CHW UNITS "-" "Friction factor"
END CLASS
CONST INTEGER MAX_PIPE = 500
PipeClass P[MAX_PIPE]
COMPONENT Pipe
PORTS
  IN Hydraulic h_in "Hydraulic inlet port"
  OUT Hydraulic h_out "Hydraulic outlet port"
DATA
  REAL L = 600 UNITS "m" "Pipe length"
  REAL D = 0.5 UNITS "m" "Pipe inside diameter"
  REAL CHW = 120. UNITS "-" "Hazen Williams Coefficient"
DECLS
  INTEGER ipipe = 0 --pipe identifier
INIT PRIORITY 100
IF (ipipe == 0) THEN
  npipe = npipe + 1
  ipipe = npipe
  IF (h_in.inode == 0) THEN
    node = node + 1
    h_in.inode = node
  END IF
  IF (h_out.inode == 0) THEN
    node = node + 1
    h_out.inode = node
  END IF
  P[ipipe].nd1 = h_in.inode
  P[ipipe].nd2 = h_out.inode
  P[ipipe].L = L
  P[ipipe].D = D
  P[ipipe].CHW = CHW
END IF
...
END COMPONENT
    
```

Desarrollo de la Librería HYDRAULIC_ST

FORMULACIÓN

✓ Sistema de Ecuaciones

- Conservación masa o flujo en los nodos
 - Caída de altura en las tuberías
 - Altura en nodos conectados a tanques
- $N^{\circ} \text{ecuaciones} = N^{\circ} \text{ nodos} + N^{\circ} \text{ tuberías} + N^{\circ} \text{ tanques}$

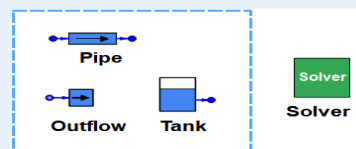
✓ Incógnitas

- Altura en los nodos
 - Flujo en las tuberías
 - Flujo de salida de los tanques
- $N^{\circ} \text{incógnitas} = N^{\circ} \text{ nodos} + N^{\circ} \text{ tuberías} + N^{\circ} \text{ tanques}$

✓ Método Newton-Raphson

✓ Uso de las Clase de la Librería SPARSE

PALETA DE COMPONENTES



Líneas de Código de la Librería

	Aproximación Clásica	Aproximación Nueva	Ratio
Líneas de Código	56	285	5

Comparación HYDRAULIC_ST con Aproximación Clásica

Caso Prueba 1

6 tuberías,
1 tanque,
3 flujos,
5 nodos
12 ecuaciones

	Aproximación Clásica	Aproximación Nueva	Ratio
Nº Ecuaciones Implícitas	3	12	4
Tiempo de CPU (ms)	0.082	0.168	2.05
Robustez	Mala	Buena	

Caso Prueba 2

716 tuberías,
4 tanques,
12 flujos,
396 nodos
1116 ecuaciones

	Aproximación Clásica	Aproximación Nueva	Ratio
Nº Ecuaciones Implícitas	328	1116	3.4
Tiempo de CPU (ms)	369	24.1	0.065
Robustez	Mala	Buena	

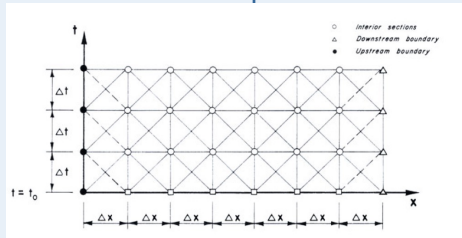
25

4. EL PROBLEMA DE LA SIMULACIÓN TRANSITORIA

26

Herramientas Específicas de Transitorios Hidráulicos:

- Utilizan el Método de las Características (MOC). Es muy preciso y consume muy poco tiempo en el caso de maniobras rápidas.

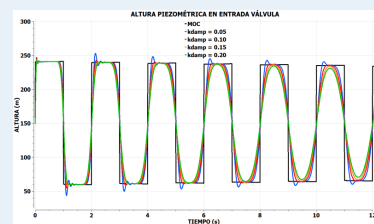
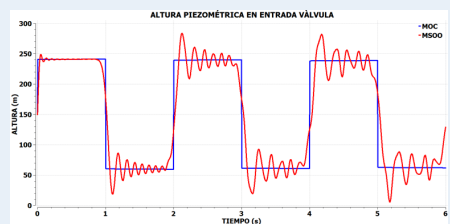
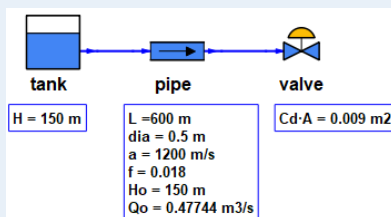
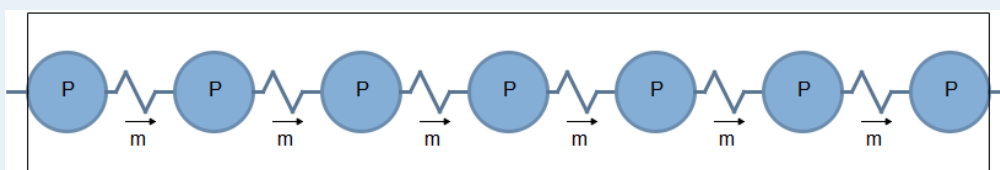


- La nodalización (división de las tuberías en nodos) se realiza de forma automática. El usuario solo define el número de nodos en la tubería más corta.
- Calculan el estacionario de una forma eficaz, considerando solo dos ecuaciones por tubería.

27

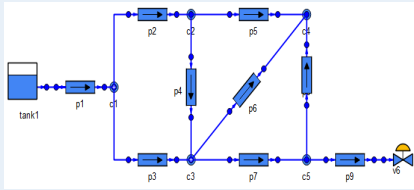
Modelado Convencional MSOO para Transitorios Hidráulicos:

Dividen la tubería como una sucesión de volúmenes y uniones, el número de nodos es dato y debe ser proporcionado para cada tubería:



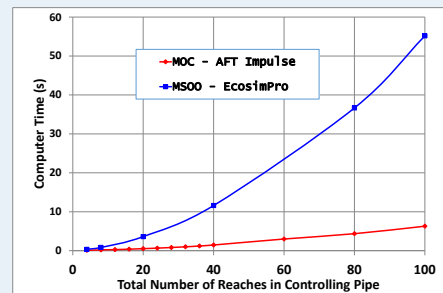
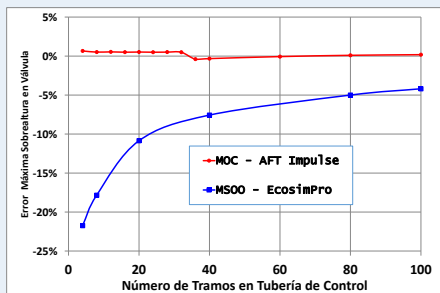
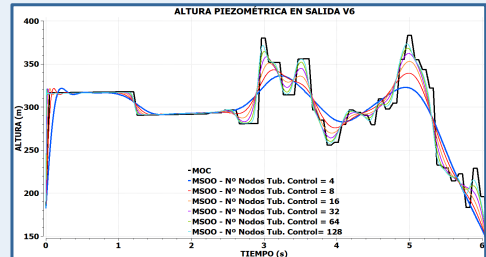
28

Método de las Características versus MSOO Convencional con EcosimPro:



Tubo No	Longitud m	Diámetro m	Factor Fricción	Velocidad Onda m/s
p1	609.60	0.91440	0.030	1005.84
p2	914.40	0.76200	0.028	1143.00
p3	609.60	0.60960	0.024	1219.20
p4	548.64	0.45720	0.020	914.40
p5	457.20	0.45720	0.020	1143.00
p6	487.68	0.45720	0.025	975.36
p7	670.56	0.76200	0.040	957.07
p8	457.20	0.60960	0.030	914.40
p9	609.60	0.91440	0.024	1005.84

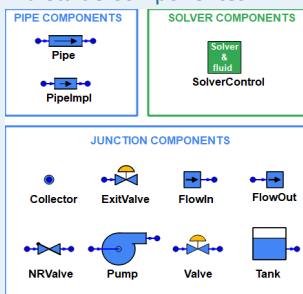
$$Q_{V6} = \begin{cases} 0.8495 \text{ m}^3/\text{s} & \text{Time} \leq 10^{-3} \text{ s} \\ 0 & \text{m}^3/\text{s} & \text{Time} > 10^{-3} \text{ s} \end{cases}$$



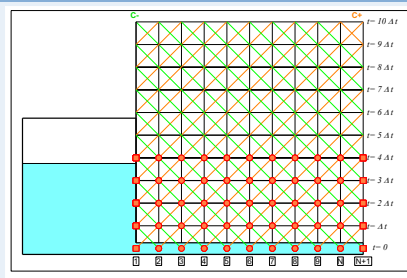
29

Desarrollo de la Librería LijHammer_ST

Paleta de Componentes



Método de las Características (MOC)



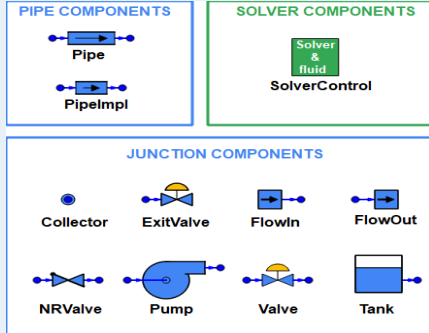
Problemas para una implementación eficiente del MOC:

1. Cálculo de secciones o nodos internos se efectúa de forma discreta con un intervalo de tiempo muestreado
2. En las condiciones de contorno, las características se trazan de manera continua, para ello es preciso interpolar en el espacio.
3. Se emplea un incremento de tiempo común para todas las tuberías

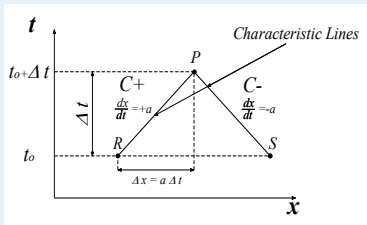
30

Desarrollo de la Librería LiqHammer_ST

Paleta de Componentes



- ✓ Método de las Característica (MOC)
- ✓ Modelo de Cavitación Discreta
- ✓ Componente Pipe Implícito para tuberías muy cortas



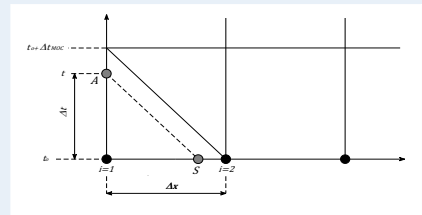
$$H_P^{t+\Delta t} = CP - BP Q_P^{t+\Delta t}$$

$$H_P^{t+\Delta t} = CM + BM Q_P^{t+\Delta t}$$

$$H_P^{t+\Delta t} = \frac{CP \cdot BM + CM \cdot BP}{BP + BM}$$

$$Q_P^{t+\Delta t} = \frac{CP - CM}{BP + BM}$$

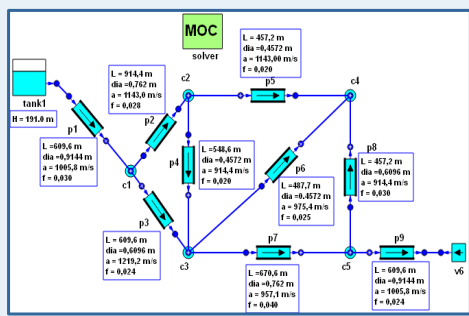
En las condiciones de contorno, las características se trazan de manera continua.



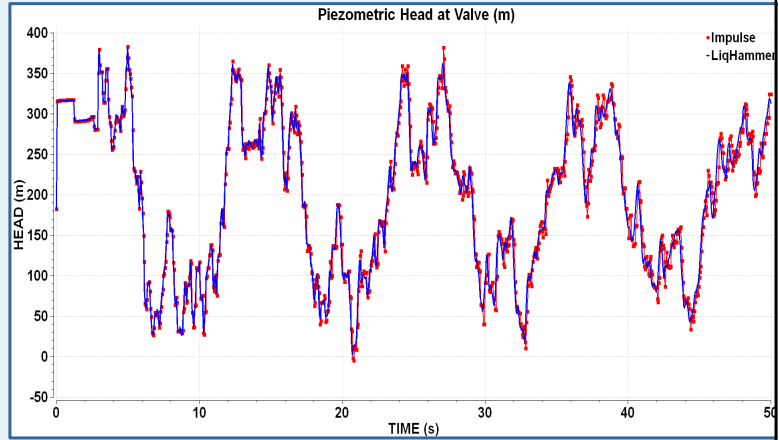
Se requiere utilizar un incremento de tiempo común para todas las tuberías

Caso de Prueba 1 de la Librería LiqHammer_ST

Modelo Caso Prueba 1



Resultados Caso Prueba 1



Desafío: Implementación del MOC con MSOO

33/35

Comparación LiqHammer_ST con Herramienta Simulación Específica de Transitorios Hidráulicos

Comparación

Para el caso 1 se ha comparado la nueva librería contra un programa comercial de golpe de ariete (AFT Impulse) variando el número de nodos.

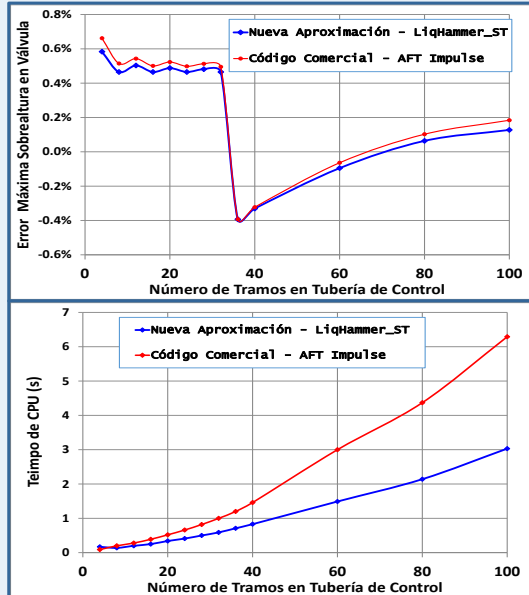
Se ha comparado:

- El error de cálculo en sobrepresión en la válvula
- El tiempo de CPU

Resultados

Los errores son casi idénticos con ambas aproximaciones, lo cual es esperable al utilizar ambos programas el mismo método de cálculo.

Los tiempos de CPU son ligeramente mejores con la nueva aproximación para el caso seleccionado.



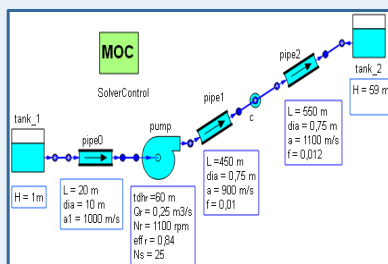
33

Desafío: Implementación del MOC con MSOO

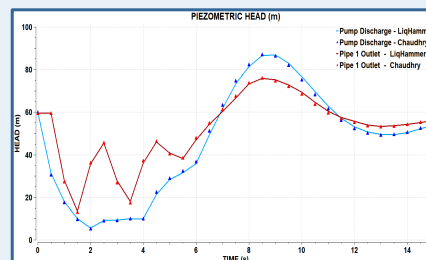
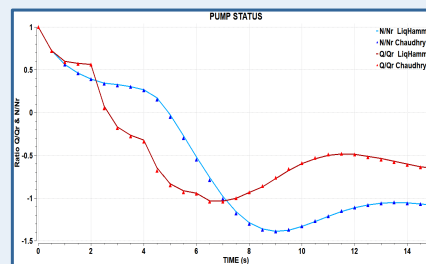
34/35

Caso de Prueba 2 de la Librería LiqHammer_ST

Modelo Caso Prueba 2



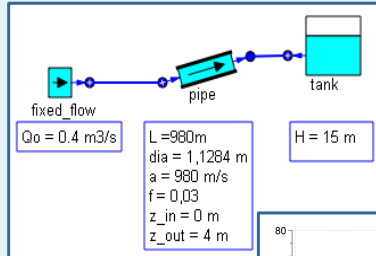
Resultados Caso Prueba 2



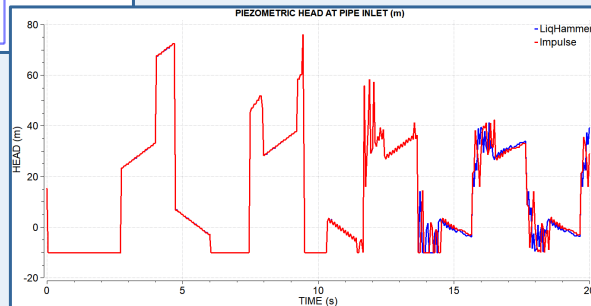
34

Caso de Prueba 3 de la Librería LiqHammer_ST

Modelo del Caso Prueba 3: Cavitación por detención flujo de entrada



Resultados Caso Prueba 3



PREGUNTA BÁSICA DE LA INVESTIGACIÓN

¿Es viable desarrollar aplicaciones de MSOO para campos específicos con la misma eficacia y facilidad de uso que los programas de simulación para tales campos?

PREGUNTA SECUNDARIA DE LA INVESTIGACIÓN

¿Se pueden implementar métodos de solución diferentes de los DAE's utilizando el lenguaje de simulación?

- ❑ Para la simulación estacionaria se ha propuesto un esquema que permite implantar cualquier método de solución con EcosimPro.
- ❑ Para la simulación transitoria, se ha propuesto un esquema que ha permitido implementar el MOC.
- ❑ Las librerías que se han presentado son librerías académicas, su fin es ilustrar la nueva aproximación. Una línea de trabajo futura es el desarrollo de librerías profesionales para:
 - Hidráulica, combinando estacionarios y transitorios en la misma librería,
 - Balances térmicos
 - Transitorios termo-fluido dinámicos en sistemas de tuberías transportando gases.



GRACIAS