

Memoria PFC – Dpto ESAII
Edmundo Guerra Paradas

Títol: Simulación de Modelos de Sistemas Dinámicos para Sostenibilidad

Volum: 1/1

Alumne: Edmundo Guerra Paradas

Director/Ponent: Antoni Grau Saldes /Yolanda Bolea Monte

Departament: ESAII

Data: 29 de Gener de 2009

DADES DEL PROJECTE

Títol del Projecte: Simulación de Modelos de Modelos Dinámicos de Sistemas para Sostenibilidad

Nom de l'estudiant: Edmundo Guerra Paradas

Titulació: Enginyeria Informàtica

Crèdits: 37.5

Director/Ponent: Antoni Grau Saldes / Yolanda Bolea Monte

Departament: ESAII

MEMBRES DEL TRIBUNAL (nom i signatura)

President: Manel Frigola Bourlon

Vocal: Rubén González García

Secretari: Antoni Grau Saldes

QUALIFICACIÓ

Qualificació numèrica:

Qualificació descriptiva:

Data: 29 de Gener de 2009

Agradecimientos

Quiero dar las gracias a Antoni Grau y Yolanda Bolea, tanto por darme la oportunidad de hacer este proyecto, como por su inestimable guía y ayuda, sin la cual creo que no habría sido posible completarlo. Asimismo me gustaría agradecer a Francisco Esquembre su compromiso para resolver dudas y problemas con EJS.

También me gustaría dar las gracias a mis padres y amigos, por su apoyo durante la realización del proyecto y aguantarme tantas cosas que yo no se si aguantaría.

Por último, también quiero dar las gracias a toda esa gente que no conozco, pero está luchando para que el legado que dejemos a generaciones venideras sea un mundo vivo de oportunidades, y no el erial marchito que estamos creando.

Edmundo Guerra Paradas

Índice

1. EL PROYECTO	- 9 -
1.1. DEFINICIÓN Y FINALIDAD.....	- 9 -
1.2. OBJETIVOS DEL PROYECTO	- 9 -
1.2.1. Elección y familiarización con el entorno de trabajo.....	- 9 -
1.2.2. Selección del conjunto de sistemas a simular.....	- 10 -
1.2.3. Estudio e implementación de los modelos matemáticos	- 10 -
1.2.4. Diseño e implementación visual de la simulación.....	- 10 -
1.2.5. Descripción de los sistemas y modelos.....	- 10 -
1.3. PLANIFICACIÓN DEL PROYECTO	- 10 -
1.4. ESTUDIO ECONÓMICO	- 12 -
2. SOSTENIBILIDAD Y SOCIEDAD	- 15 -
2.1. ¿SOSTENIBILIDAD O SUSTENTABILIDAD?	- 15 -
2.2. REPASO DE HISTORIA DE LA SOSTENIBILIDAD.....	- 16 -
2.3. TIC, SOCIEDAD DE LA INFORMACIÓN Y SOSTENIBILIDAD.....	- 18 -
3. MODELOS, SIMULACIONES Y REQUISITOS.....	- 20 -
3.1. SELECCIÓN DE MODELOS	- 20 -
3.1.1. Modelado para predicción de Nivel Sónico Equivalente, Leq.....	- 20 -
3.1.2. Modelado de un secador solar de sultaninas	- 20 -
3.1.3. Modelado de autodepuración de aguas de emisario submarino en campo lejano.....	- 21 -
3.1.4. Simulador ecológico PZNPO	- 21 -
3.1.5. Del orden al caos y viceversa en ecosistemas acuáticos.....	- 21 -
3.1.6. Modelado de flujo de dos fases de drenaje de una veta de carbón.....	- 21 -
3.1.7. Modelado del flujo de metano en los Everglades de Florida	- 21 -
3.2. REQUISITOS COMPLEMENTARIOS.....	- 22 -
4. SOFTWARE DE SIMULACIÓN.....	- 24 -
4.1. EASY JAVA SIMULATIONS Y SU ELECCIÓN.....	- 24 -
4.2. ALTERNATIVAS A EJS ESTUDIADAS.....	- 25 -
4.2.1. MATLAB + Simulink + GUIDE.....	- 25 -
4.2.2. Vensim PLE.....	- 25 -
4.2.3. Stella & Netsim.....	- 26 -
4.3. ¿COMO FUNCIONA EJS?	- 26 -
4.4. ELEMENTOS VISUALES EMPLEADOS	- 29 -
4.4.1. Elementos de control	- 30 -
4.4.2. Elementos de visualización.....	- 31 -
5. MODELADO PARA PREDICCIÓN DE NIVEL SÓNICO EQUIVALENTE, LEQ	- 34 -
5.1. INTRODUCCIÓN.....	- 34 -
5.2. DESCRIPCIÓN DEL MODELO MATEMÁTICO	- 34 -
5.3. ASPECTOS TÉCNICOS.....	- 37 -
5.4. VISUALIZACIÓN Y CONTROL DE LA SIMULACIÓN	- 38 -
6. MODELADO DE UN SECADOR SOLAR DE SULTANINAS.....	- 41 -
6.1. INTRODUCCIÓN.....	- 41 -
6.2. DESCRIPCIÓN DEL MODELO MATEMÁTICO	- 42 -
6.3. VISUALIZACIÓN Y CONTROL	- 45 -
7. SIMULADOR ECOLÓGICO PZNPO.....	- 47 -
7.1. INTRODUCCIÓN.....	- 47 -
7.2. DESCRIPCIÓN DEL MODELO MATEMÁTICO	- 47 -

7.3.	VISUALIZACIÓN Y CONTROL	- 52 -
8.	MODELADO DE AUTODEPURACIÓN DE AGUAS DE EMISARIO SUBMARINO EN CAMPO LEJANO.....	- 55 -
8.1.	INTRODUCCIÓN.....	- 55 -
8.2.	DESCRIPCIÓN DEL MODELO MATEMÁTICO	- 55 -
8.3.	CONTROL Y VISUALIZACIÓN.....	- 59 -
9.	DEL CAOS AL ORDEN Y VICEVERSA EN ECOSISTEMAS ACUÁTICOS	- 62 -
9.1.	INTRODUCCIÓN.....	- 62 -
9.2.	DESCRIPCIÓN DEL MODELO MATEMÁTICO	- 62 -
9.3.	CONTROL Y VISUALIZACIÓN	- 65 -
9.4.	RESPUESTAS FUNCIONALES DE HOLLING	- 68 -
9.5.	FASES DE ESTADO DEL SISTEMA	- 70 -
10.	MODELADO DE FLUJO DE DOS FASES DEL DRENAJE DE UNA VETA DE CARBÓN	- 73 -
10.1.	INTRODUCCIÓN.....	- 73 -
10.2.	DESCRIPCIÓN DEL MODELO MATEMÁTICO	- 73 -
10.3.	VISUALIZACIÓN Y CONTROL	- 78 -
11.	MODELADO DEL FLUJO DE METANO EN LOS EVERGLADES DE FLORIDA.....	- 82 -
11.1.	INTRODUCCIÓN.....	- 82 -
11.2.	DESCRIPCIÓN DEL MODELO MATEMÁTICO	- 82 -
11.3.	ASPECTOS TÉCNICOS.....	- 91 -
11.4.	CONTROL Y VISUALIZACIÓN	- 93 -
12.	DOCUMENTACIÓN Y DISTRIBUCIÓN DE LAS SIMULACIONES	- 96 -
12.1.	DOCUMENTACIÓN DE LAS SIMULACIONES	- 96 -
12.2.	DISTRIBUCIÓN DE LAS SIMULACIONES.....	- 97 -
13.	CONCLUSIONES	- 100 -
	BIBLIOGRAFÍA Y REFERENCIAS	- 102 -

1. El Proyecto

1.1. Definición y finalidad

El principal objetivo de este PFC es la realización de una serie de simulaciones de sistemas dinámicos de temática vinculada al medio ambiente y la sostenibilidad, basándose en los modelos matemáticos que los describen. Estos modelos matemáticos se implementan mediante *Easy Java Simulations*, una herramienta enmarcada dentro del proyecto *Open Source Physics*, así como otras bibliotecas de *Java*.

Las simulaciones serán representadas visualmente como *Applets*. Estos pueden ser tanto ejecutados individualmente como programas o mostrarse dentro de páginas HTML vía navegador –siempre que se cuente con soporte para la máquina virtual de *Java*–.

La finalidad de realizar estas simulaciones es de carácter principalmente educativo y divulgativo, dando a conocer sistemas y modelos que describen aspectos ecológicos o ambientales de interés, así como demostrar el potencial del programario utilizado como herramienta de modelado matemático y visual.

1.2. Objetivos del proyecto

Con el fin de realizar el proyecto bajo una estrategia mejor dirigida, se han definido una serie de objetivos menores. Estos están relacionados con la planificación y tareas del proyecto estrechamente, y en cierta manera, marcan las pautas empleadas para acotar las fases del proyecto. A saber:

1.2.1. Elección y familiarización con el entorno de trabajo

Easy Java Simulations es una potente herramienta basada en *Java*, la cual permite emplear todas las características de la programación en *Java*. Entre ellas el uso de bibliotecas basadas en *Java*, un detalle relevante cuando hablamos de modelos matemáticos con cálculos probablemente complejos. Aún así, permite programar ecuaciones diferenciales ordinarias y controlar el flujo de ejecución de forma relativamente sencilla. No obstante, se evaluarán las posibles alternativas, ya que no

se puede descartar que alguna de ellas resulte más adecuada y por tanto sea la opción óptima.

1.2.2. Selección del conjunto de sistemas a simular

Partiendo de una selección inicial de dos sistemas y sus modelos, se han ido escogiendo y descartando modelos candidatos en función del posible interés, complejidad matemática y técnica, y la viabilidad de ser modelados con la documentación disponible sobre ellos. Aún así no es raro que un extenso *paper* describiendo minuciosamente el modelo de un sistema cometa omisiones o contenga contradicciones, y que esto solo sea descubierto una vez se procede a la fase de implementación y testeo.

1.2.3. Estudio e implementación de los modelos matemáticos

Una vez un sistema es escogido para simularlo, se pasa a estudiar el modelo matemático que lo representa, para ser implementado en EJS. Con frecuencia, a la hora de implementar el modelo matemático, se toman decisiones importantes de diseño que condicionan el desarrollo del aspecto visual de la simulación.

1.2.4. Diseño e implementación visual de la simulación

El desarrollo del aspecto visual es un punto crítico de la simulación. Por muy perfecto que sea el modelo que describe el funcionamiento del sistema, solo será perceptible en función de los *Applets* en los que se distribuya y visualice la representación escogida para los datos. Esta visualización ha de poder verse en navegadores estándar –incluso a resoluciones bajas- sin suponer una gran molestia.

1.2.5. Descripción de los sistemas y modelos

Dada la finalidad educativa y divulgativa de las simulaciones, es necesario describir con precisión y claridad el sistema y como ha sido modelado. También, de ser necesario, se detallarán explicaciones sobre conceptos oscuros. El factor más deseable es conseguir hacerlo comprensible a gente no experta en las distintas áreas de conocimiento introducidas.

1.3. Planificación del proyecto

El proyecto está pensado para realizarse a lo largo de quince semanas. Esto viene a suponer unas 600 horas de trabajo según la normativa habitual de proyectos.

Teniendo en cuenta la descripción hecha en los objetivos del proceso de trabajo, la planificación será iterativa respecto a diversas fases, que se irán repitiendo, con el fin de aprovechar la retroalimentación y el conocimiento adquirido mediante la experiencia.

De este modo, una vez confirmada la elección de EJS como herramienta de desarrollo principal, y adquirida una serie de pericias básicas en su uso, se iniciará la parte productiva del proyecto. A lo largo de este segmento del proyecto, las etapas correspondientes a los objetivos de selección de modelos, estudio e implementación de los modelos, diseño e implementación de vistas y descripción de modelos irán sucediéndose cíclicamente, realizando una iteración por modelo.

Una vez completado el conjunto de modelos seleccionados, la tarea final consistirá en la recopilación y redacción final de las anotaciones y descripciones que compondrán la memoria del proyecto, tal y como muestra la (figura 1).

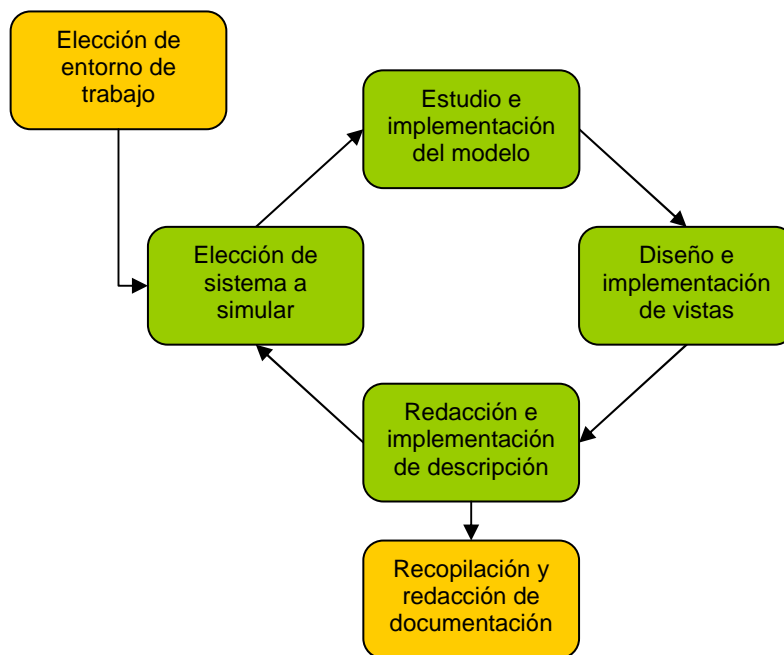


Fig. 1: Diagrama de procesos del proyecto

Siendo así, inicialmente se desconoce la carga de trabajo y tiempo proyectado para cada sistema a modelar, pues de hecho se desconocen los propios sistemas que serán escogidos.

Por este motivo se muestra a continuación la planificación final real de trabajo. Cada una de las fase que forman el ciclo de desarrollo, pese a realizarse en hasta 7 iteraciones, se ha considerado una única tarea por simplicidad. Se observa como el

proceso de elección de modelo se realiza en menos ocasiones, ya que en una de ellas fueron seleccionados dos modelos simultáneamente.

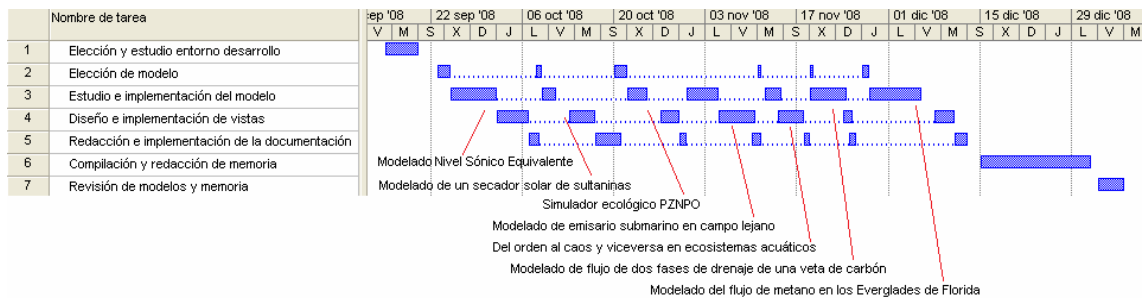


Fig. 2: Diagrama de Gantt del proyecto

1.4. Estudio económico

Realizar un estudio económico de un proyecto de estas características resulta cuanto menos complejo y arduo. Trabajar con nuevas tecnologías, y sobretudo, en áreas de conocimiento totalmente desconocidas, como ecología y bioquímica, introduce una incertidumbre difícil de cuantificar.

No obstante el estudio de los costes asociados a recursos no humanos es enteramente calculable, así como realizar una estimación aproximada de los costes humanos por analogía, comparando los presuntos requisitos y el volumen de trabajo esperado. De esta forma, podremos aproximar la repartición de la carga de trabajo, estimable en alrededor de 600 horas, entre los distintos roles asignados.

Costes de recursos no humanos:

- Licencias de programario:
- Sistema Operativo: 0 €
 - *Linux*
- Entorno de desarrollo: 0 €
 - *Easy Java Simulations*
- Programario de ofimática 0 €
 - *Open Office*
- Equipo de desarrollo: 0 €/ 600 €

- *PC doméstico convencional*

Costes de recursos humanos:

- Director de proyectos 3500 €
 - 70 Horas a 50 €
- Analista 5250 €
 - 150 Horas a 35 €
- Programador 7600 €
 - 380 Horas a 20 €

Total 16350 €/ 16950 €

El estudio económico considera que se emplean las opciones más económicas en materia de programario, siempre que cumplan con los requisitos mínimos. El equipo de desarrollo, se contabiliza tanto como un coste de adquisición del proyecto, como suponiendo que se disponga previamente de éste y no suponga coste alguno. En cuanto a los roles de los distintos recursos humanos, se han empleado costes relativamente bajos en los cálculos, para estimar precios más cercanos al mercado autónomo, que no al de la gran consultoría de prestigio –con un modelo de negocio con sus ventajas e inconvenientes, pero con unos costes generalmente excesivos-.

Los roles interactúan con el proyecto de forma habitual. El director de proyectos se responsabiliza de la gestión y planificación, en la medida que éstas se realizan. El analista es la pieza más importante, ya que a diferencia de otros proyectos, el éxito depende enteramente de él, al ser el responsable de estudiar los modelos, y diseñar como se implementarán en EJS. El programador será el verdadero experto en EJS, teniendo la experiencia de primera mano en su uso, y brindando al analista la retroalimentación necesaria para que éste pueda depurar y adaptar mejor sus diseños a las posibilidades del entorno. En cuanto a la documentación se asume que para cada parte o área de ésta, la realizará la persona cualificada de menor coste por hora. Así el grueso de las labores de documentación recaerá sobre el programador y el analista.

Resulta interesante estudiar la desviación producida respecto a la distribución de carga de trabajo típica, además de la habitual desviación en volumen de horas de trabajo. Así, mientras el rol programador ha visto su volumen de trabajo real reducido,

gracias sobretodo a la potencia de EJS como herramienta de programación, el rol analista ha visto su carga de trabajo incrementada. Esto se ha debido sobretodo a la necesidad de adquirir amplios conocimientos en áreas de conocimiento nuevas y una labor de documentación no prevista, para hallar datos erróneos e incompletos de los artículos en que se basan los modelos. Así, a nivel de recursos humanos, tenemos un nuevo cálculo de costes, dándonos un nuevo total:

Costes de recursos no humanos: 0/600 €

Costes de recursos humanos:

- Director de proyectos 3500 €
 - 70 Horas a 50 €
- Analista 6650 €
 - 190 Horas a 35 €
- Programador 6800 €
 - 340 Horas a 20 €

Total 16950 €/ 17550 €

2. Sostenibilidad y Sociedad

2.1. ¿Sostenibilidad o sustentabilidad?

Tanto sostenibilidad como sustentabilidad –están aceptadas por la comunidad científica como sinónimas-, en un contexto ecológicamente amplio, se refieren a la habilidad de un ecosistema para mantener los procesos y funciones ecológicas, así como su biodiversidad y productividad en el futuro. Aunque con más frecuencia se refiere concretamente a la relación entre el ser humano, su actividad, y como estos se relacionan con dicha capacidad de perdurar y mantenerse, como una especie de equilibrio.

El término en si es lo bastante amplio que puede ser aplicado tanto a formas de vida y organización biótica, ecosistemas, y otras facetas de la vida en la Tierra como a actividades, ciencias y disciplinas estrictamente humanas. En cierta forma, el estudio de la sostenibilidad podría englobar cualquier casi disciplina de estudio humana, pues conceptualmente la sostenibilidad contemplaría aspectos de la relación entre individuos y la sociedad, como derechos, libertad, y justicia de esta relación –justicia en términos éticos más que legales-.

Casi siempre se asocia la sostenibilidad al desarrollo sostenible, ya que para que un desarrollo sea sostenible, éste ha de poseer cierta sostenibilidad. Normalmente se acepta como tal aquel desarrollo que “satisface las necesidades de las generaciones presentes, sin comprometer las posibilidades de las del futuro para atender sus propias necesidades”, definido por el Informe Brundtland de la ONU en 1987. Por supuesto, dadas las enormes implicaciones sociales y económicas del desarrollo sostenible, no hay una aceptación unánime en cuanto al significado del término y su aplicación.

Cabe matizar que la sostenibilidad referida al desarrollo sostenible incluye no solo el desarrollo ecológico, sino también el desarrollo social y económico, en cuanto a que se han de “satisfacer las necesidades”. Esto es especialmente importante para países en vías de desarrollo: por muy buen rendimiento medioambiental que ofrezca, una planta o instalación de producción puede ser poco o nada sostenible si sus costes de operación son excesivos para una economía débil. Además se añade una dimensión política al desarrollo sostenible, donde distintas corrientes pugnan por adueñarse de

los réditos electorales de éste, mientras otros lo intentan desterrar de las preocupaciones de la sociedad en alas de preservar los intereses político-económicos de los que participan.

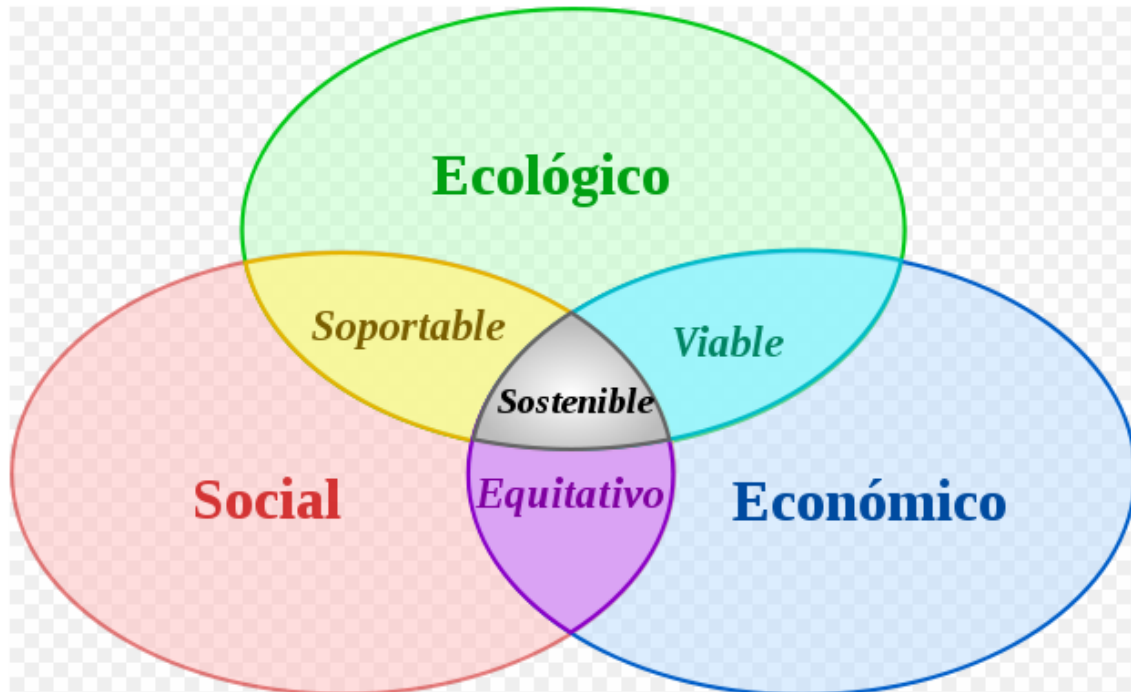


Fig. 3: Esquema de criterios, valores y tipologías de desarrollo sostenible

Uno de los grandes problemas de la sostenibilidad viene dado por el anhelo de igualdad y justicia de la humanidad: los países actualmente en vías de desarrollo tienen tanto derecho como los países desarrollados a crecer económica y socialmente. No obstante, el crecimiento de los países ya desarrollados se hizo a expensas de grandes daños ambientales y ecológicos. Y más allá de los posibles daños que supondría que ciertas potencias alcanzasen un nivel de desarrollo avanzado, está el simple hecho de que la humanidad hace tiempo que superó el umbral de la sostenibilidad.

2.2. Repaso de historia de la sostenibilidad

A lo largo de la historia, el éxito y prosperidad de las civilizaciones humanas se ha medido casi siempre en función del desarrollo cultural y económico de éstas. Desde la opulencia de los imperios de la antigüedad, pasando por la inabarcable extensión de la España donde no se ponía el Sol, hasta la salvaje globalización económica actual, la mayoría de civilizaciones han medido su éxito en términos sociales y económicos.

En este supuesto estamos de suerte y en racha: desde el siglo XIX, iniciado por la revolución industrial, un imparable crecimiento económico sostenido ha permitido un desarrollo social admirable. Lamentablemente, este crecimiento sostenido de la civilización no ha sido ejemplarmente sostenible. De hecho, exceptuando algunos visionarios e iluminados con sus teorías a la catástrofe maltusiana, no es hasta la década de los 60 del siglo XX que empieza a haber alguna preocupación manifiesta por el desarrollo sostenible.

El Club de Roma, formado en 1968 por personalidades como premios Nobel y jefes de estado, encargó al MIT un estudio que sería publicado en 1972, “Los límites del crecimiento”, antes de la primera crisis del petróleo. Tanto este primer informe como su actualización “Más allá de los límites del crecimiento”, en 1992, arrojaron la misma conclusión: la humanidad ha superado con creces la capacidad de carga del planeta. El crecimiento económico y demográfico humano excede la capacidad de la Tierra, consumiendo sus recursos y deteriorando el medio ambiente más rápido de lo que el planeta puede recuperarse.

No obstante, pese a saberse que el comportamiento de parte de la humanidad era totalmente negligente, durante muchos años no se hizo nada, escudándose los líderes responsables de estos excesos en las más diversas excusas: la responsabilidad con los accionistas, responsabilidad con el bienestar de los ciudadanos, responsabilidad en la lucha contra el enemigo de turno... Donde la única responsabilidad que había era la disuelta en el efecto grupo y el compromiso con los intereses económicos.

Solo cuando la evidencia ha sido totalmente innegable, y la posible factura a medio y largo plazo de seguir destruyendo el medio ambiente global excesivamente cara, empieza a haber un aparente movimiento entre los responsables para buscar soluciones. Tal vez el hecho de que un ex vicepresidente estadounidense haya conseguido un premio Nobel de la Paz por un documental conjunto con un grupo de trabajo de la ONU, dando sus 15 minutos de fama al calentamiento global, el medio ambiente y la sostenibilidad, puede haber tenido algo que ver en el aparente resurgir del ecologismo, aunque solo sea como una moda.

Por que no nos engañemos. Nadie sabe lo que es la sostenibilidad. Es una palabra que suena muy bien, sobretodo en términos como desarrollo o crecimiento sostenible - aunque este último sea un oxímoron-, pero que no existe. Al menos, no existe oficialmente para la RAE, o no tiene la suficiente importancia y difusión como para ser incluida a la lengua castellana. Todo lo contrario que *Pen Drive*, un burdo anglicismo, pero que parece que si que tiene la importancia y difusión como para ser inclusión

confirmada en la próxima edición de su diccionario. Quedan claras las preocupaciones del ciudadano medio del estado español, o al menos que el desarrollo sostenible no es una de ellas.

2.3. TIC, Sociedad de la Información y Sostenibilidad

Desde una óptica estrictamente ecologista, un proyecto sobre sostenibilidad en la carrera de informática puede ser un poco hipócrita: la industria informática es de las más insostenibles que se puedan concebir. A nivel de mercado no es raro que sus equipos se consideren obsoletos en periodos de incluso menos de un año, y poco más que basura inservible a sustituir en menos de 3 años, aunque la gente siga haciendo lo mismo con ellos que hace media década: trabajar lo indispensable y dedicarse a perder tiempo por Internet. Peor puede resultar aún si se consideran los componentes empleados en la producción de equipos: su toxicidad es tal que se omiten nombres por evitar riesgos asociados a su mera lectura. Por mucha presión que puedan ejercer campañas como *Greener Electronics* de *Greenpeace*, todavía queda mucho para que en un equipo informático lo más verde deje de ser la placa base o la pátina del cobre.

Aún así, es innegable el potencial que poseen las nuevas tecnologías de la comunicación y la información, incluso por encima de la capacidad del hombre para rebajar la dignidad humana, el cual ha permitido el desarrollo de la llamada *Sociedad de la Información*. En esta sociedad, pese al problema de la brecha digital, hipotéticamente todo el mundo puede participar de ella, produciendo y adquiriendo información y conocimiento.

Más allá de las implicaciones económicas de esta sociedad, con sus cambios en los modelos tanto productivos como de gestión, lo que nos interesa es el potencial de sensibilización y divulgación. A través de Internet, cualquier individuo –de países sin censura digital- puede participar de la sociedad de la información, adquiriendo y produciendo conocimiento de cualquier materia que le resulte interesante. Esto cambia radicalmente la percepción del individuo del mundo que le rodea, liberándolo a un universo de conocimientos y opiniones diversos –aunque no por ello veraces-. Donde antaño alguien que sintiese no satisfecha su curiosidad con la prensa o los noticiarios respecto cualquier tema, veía sus opciones de investigación limitada, ahora su problema será cribar o distinguir que es lo verdaderamente interesante en el mar de información vomitado por su navegador de Internet. Y esa es la promesa de esperanza de la sociedad de la información, que los individuos, enfrentados a mares de

información y evidencias, empiecen a percibir un poco mejor el mundo que los rodea e intenten comprender los problemas que les amenazan.

No obstante, cabe recordar que la sostenibilidad y el medio ambiente son términos esencialmente malinterpretados por la sociedad: estigmatizados como cosas de *hippies* y progresistas trasnochados por muchos, e indefinibles por muchos otros – como la RAE-. Y la frivolidad con la que son usado como herramientas de RRPP impunemente por partidos políticos, artistas, y cualquiera que crea poder aumentar su popularidad o base de apoyo subiéndose al carro verde, tampoco ayuda a que mucha gente deje de considerarlos poco más que una moda recurrente, como los pantalones de campana o los cinturones anchos.

Así resulta imprescindible hacer el mayor esfuerzo posible, desde el potencial de la TIC, por educar y divulgar la importancia del desarrollo de una actividad humana más sostenible, y que la sociedad de la información despierte y luche para salvar un planeta agonizante.

3. Modelos, simulaciones y requisitos

3.1. Selección de modelos

La colección de modelos a simular ha sido escogida durante la realización del mismo proyecto. Así, partiendo de dos modelos iniciales escogidos por su relativa simplicidad, se fueron añadiendo modelos, basándose principalmente en los criterios de interés y relevancia en materia de sostenibilidad medioambiental, y creciente dificultad a nivel de implementación. Esta estrategia ha permitido una mayor retroalimentación, aprovechando la experiencia adquirida en un entorno nuevo desarrollando un producto para escoger y valorar mejor los siguientes.

Esta mayor comprensión se podrá apreciar viendo como partiendo de modelos basados en una ecuación relativamente simple, se concluyó modelos de varias decenas de ecuaciones diferenciales y de variables de estado.

Ciertamente los temas más recurrentes han sido las aguas, tanto dulces como saladas, y la producción y liberación de metano. Comentar la importancia de las aguas a nivel de sostenibilidad ambiental resulta un poco innecesario, pues se podría resumir que el agua es la vida del planeta y todos los seres que lo pueblan. Por otra parte, se ha considerado interesante el estudio del metano, uno de los principales gases de efecto invernadero, pero con una enorme utilidad, al poder ser empleado como fuente energética siendo un componente mayor del llamado gas natural.

3.1.1. Modelado para predicción de Nivel Sónico Equivalente, L_{eq}

Evalúa dadas unas condiciones de tráfico el nivel de molestia y ruido, y su equivalente energético en las proximidades de una vía rodada de alto tráfico.

3.1.2. Modelado de un secador solar de sultaninas

Este modelo describe el comportamiento de un sistema de secado de sultaninas para producir pasas. Conociendo variables como la temperatura ambiental y la velocidad a la que se hace circular el aire por el sistema, se puede simular la temperatura a la que serán sometidas las uvas, y como van secándose al reducirse su humedad relativa.

3.1.3. Modelado de autodepuración de aguas de emisario submarino en campo lejano

Simula el comportamiento de los contaminantes enviados al mar por medio de un emisario submarino, calculando tasas de disolución, depuración, velocidad a la que se mueve las aguas contaminadas y nivel de contaminación a una distancia determinada. Permite observar hasta que punto la luz solar depura los mares, y su influencia en la salubridad de las aguas.

3.1.4. Simulador ecológico PZNPO

Este modelo está basado en el estudio de un estanque en la UPC de Castelldefels. En función de datos introducidos por medio de bases de datos o funciones de simulación, se simula el ciclo de distintos nutrientes y componentes a través de los niveles tróficos presentes en el estanque, y se estima la emisión de metano producida por el mismo.

3.1.5. Del orden al caos y viceversa en ecosistemas acuáticos

Esta simulación muestra el comportamiento de un sistema ecológico compuesto de una red trófica de tres niveles. Muestra como en ciertas circunstancias un ecosistema puede establecer nuevos equilibrios tras superar estados caóticos, pero la más leve variación en un parámetro puede de igual forma extinguir una especie destrozando ecosistema.

3.1.6. Modelado de flujo de dos fases de drenaje de una veta de carbón

Conociendo las características del suelo y la veta a drenar, y una serie de datos de presión obtenidos mediante muestras alrededor del punto a perforar, se predice como evolucionará la presión de agua y gases en el punto de drenaje perforado y sus proximidades.

3.1.7. Modelado del flujo de metano en los Everglades de Florida

Mediante el complejo sistema descrito, se simula el comportamiento y evolución de componentes como el acetato, sulfato, oxígeno y metano, así como la forma en la que los distintos organismos interactúan con ellos en las marismas y pantanos poco profundos de Florida, tomando en consideración decenas de variables y parámetros.

3.2. Requisitos complementarios

Las simulaciones, basadas en modelos ambientales principalmente, tienen como principal fin la divulgación en materia de sostenibilidad, especialmente a nivel ambiental. De ellas se espera que cualquier persona con acceso a ciertos recursos de telecomunicaciones y educacionales, y un mínimo de interés, pueda al menos llegar a ellas y emplearlas, siendo deseable que se formase una vaga idea de que exponen sus modelos. Pero además, todo esto ha de compatibilizarse con el máximo rigor y precisión científica, a nivel de cálculos y exposición de resultados.

Esto condiciona una serie de características que han de poseer las simulaciones, en forma de requisitos complementarios, que deben de ser satisfechos en la mayor parte posible para considerar que se completan realmente los objetivos de proyecto.

Otra parte del proyecto es evaluar la herramienta empleada en distintos niveles. Uno de ellos es como simulador de modelos matemáticos. Dado que se empleará una herramienta diseñada para ellos, esta función se satisface plenamente. Por ello las simulaciones serán precisas y exactas. No obstante, también es interesante evaluarla como herramienta para la creación de las interfaces GUI necesarias. Para ello se ha de intentar obtener los mejores acabados posibles a nivel gráfico.

Dado que ya se ha especificado que el usuario objetivo de las simulaciones no se limita a miembros de la comunidad científica e ingenieros, las interfaces, y vistas han de ser lo más intuitivas y atractivas posible. Salidas visuales en forma de interminables secuencias de números son del todo inaceptables como única opción. Adicionalmente a una visualización de salida científica y precisa, hay que disponer mecanismos de salida con un significado visual intuitivo, tomando como un estándar mínimo en este concepto las gráficas de evolución y similares.

Los mecanismos de control deberían de ser simples e intuitivos. En la medida de lo posible, se han de emplear interfaces ya conocidas y codificadas socialmente, aceptadas por todo el mundo. Un buen ejemplo sería el uso de terminologías clásicas como *Play* y *Pause*, un préstamo tomado de los sistemas de reproducción en general. Asimismo, dada la integración de las simulaciones en entornos de navegación, es aconsejable que el usuario no necesite usar teclado, evitándolo del todo a ser posible. La interacción humana será preferentemente por ratón, disponiendo los elementos necesarios para que el usuario introduzca los datos oportunos de la forma más cómoda posible.

Un punto importante a tener en cuenta es la accesibilidad a estas simulaciones. Sería interesante que tuviesen los menores requisitos de sistema posibles a todos los niveles. La operabilidad inter-plataforma es deseable, dado que la red, pese a los estándares de hecho como Windows, no entiende de sistemas operativos ni navegadores. Además, se persigue que no requieran instalación ellas mismas, ni de ningún programa externo, ya que es improbable que el usuario medio de Internet interesado en ecología y sostenibilidad tenga en su equipo un potente programario matemático/científico estilo.

Asimismo, dentro del ámbito de la accesibilidad, se ha de considerar la visibilidad de las simulaciones. Pese a que actualmente es difícil encontrar equipo que opere habitualmente a resolución SVGA (800x600 píxeles), no se puede descartar esta posibilidad. Por ello se considera de gran importancia, pese a complicar enormemente la satisfacción de otros requisitos, que se pueda ver la simulación en navegadores trabajando a esta resolución de forma cómoda. Esto conlleva, entre otros puntos, el poder ver la vista (o cada una de las vistas) de la simulación completa sin tener que recurrir a deformación o exploración con barras deslizadoras, ni nada que suponga molestias innecesarias.

De este modo, los requisitos complementarios generales para las simulaciones los podemos resumir en:

- Mecanismo de control fácil e intuitivo
- Evitar el uso de teclado
- Precisión y exactitud científicas
- GUI potente y de calidad:
 - Visible en navegadores a SVGA cómodamente
 - Comprensible e intuitiva para profanos
- Mantener el rigor científico
- Pocos requisitos para el sistema cliente
- Operabilidad inter-plataforma deseable
- No necesidad de programario específico

4. Software de simulación

Para la implementación de las simulaciones de los modelos escogidos finalmente se ha empleado *Easy Java Simulations*, o también EJS.

4.1. Easy Java Simulations y su elección

Easy Java Simulations es una herramienta de modelado y de autor para la creación de simulaciones científicas con finalidades pedagógicas. Está pensado más para usuarios de perfil científico técnico en general, no en programadores profesionales. Por ello su diseño permite a sus usuarios trabajar a un alto nivel conceptual, usando un conjunto de herramientas simplificadas y concentrar la mayoría de su tiempo en los aspectos científicos de la simulación. Además muchas de las tareas simples pero tediosas están implementadas para poder hacerlas automáticamente en poco tiempo [URL01].

No obstante, el resultado final, generado automáticamente por EJS a partir de la descripción del usuario, podría pasar, en términos de eficiencia y sofisticación, por la creación de un programador profesional, especialmente en términos visuales.

EJS crea aplicaciones *Java* que son independientes y multiplataforma, llamadas *applets*, que se pueden visualizar usando cualquier navegador web -y por tanto ser distribuidos a través de Internet-. Éstos pueden leer datos a través de la red y ser controlados usando *scripts* -conjuntos de instrucciones- incluidos en las páginas HTML. El único requisito es que el sistema sobre el cual se visualicen o ejecuten los *applets* disponga de una versión lo bastante actualizada de *Java Runtime Environment*, o JRE.

Este JRE es un conjunto de utilidades que permite la ejecución de programas *Java* sobre todas cualquier plataforma, permitiendo que un mismo ejecutable *Java*, en este caso, los *applets* creados, funcionen en distintas plataformas sin volverlo a compilar. Esta característica nos da una interoperabilidad total, satisfaciendo uno de los criterios completamente. En cuanto al requisito de no requerir instalación, se podría decir que es raro que un equipo con acceso a Internet no tenga ya instalado JRE. Incluso sin acceso Internet, tanto las distintas versión de sistemas operativos de Mac, Microsoft y la mayoría de distribuciones de Linux suele llevar una versión bastante actualizada.

Gracias a todo lo expuesto, EJS consigue satisfacer plenamente los requisitos complementarios asociados a las simulaciones de los modelos, destacando enormemente entre las alternativas estudiadas. Más aún si se considera que es fruto de una iniciativa *Open Source* enmarcada dentro del proyecto *Open Source Physics*. Así supone un coste cero, a todos los niveles, tanto de adquisición de licencias, como mantenimiento, actualización y soporte técnico.

En este último aspecto, el soporte técnico, EJS ofrece una última ventaja. EJS ha sido creado por Francisco Esquembre en la Universidad de Murcia. Esto conlleva que haya una profusa comunidad de trabajo en el estado, tanto en el desarrollo como el uso de EJS, de la que él mismo participa activamente ofreciendo ayuda y soporte.

4.2. Alternativas a EJS estudiadas

La elección de EJS como herramienta de desarrollo ha sido por su óptima adecuación a nuestras necesidades. Aún así, cabe mencionar que otras alternativas han sido estudiadas para su posible uso. A continuación se comentan algunas de las alternativas más destacadas.

4.2.1. *MATLAB + Simulink + GUIDE*

MATLAB, de *MATrix LABoratory*, es uno de los más potentes programarios matemáticos comerciales desde hace años. Ofrece un entorno de desarrollo integrado, con un lenguaje de programación propio, el *M*. El paquete incluye las herramientas *Simulink* y *GUIDE*, que son una plataforma de simulación multidominio y un editor de interfaces gráficas de usuario respectivamente [URL02].

MATLAB habría sido una buena alternativa, pero su lenguaje propio y complejidad de uso hubiera implicado un tiempo de aprendizaje excesivo, restándole atractivo a la alternativa que representa. Además, *MATLAB* es un software propietario de alta calidad y un coste económico relativamente elevado.

4.2.2. *Vensim PLE*

Vensim es una herramienta para el desarrollo, análisis y empaquetado de modelos dinámicos, creado por Ventana Systems. Como indicador de calidad del mismo se puede citar que es parte del programario utilizado con fines docentes y de investigación en la Sloan School of Management del MIT (Massachusetts Institute of Technology). Además, pese a ser un programa comercial, existe una versión gratuita

para usos docentes y estudiantes, Vensim PLE. Esta versión, Vensim PLE, dispone a nivel de modelado de muchas más opciones que EJS, pero éstas no son ampliables con librerías como EJS. Pese a ser una versión de estudiantes, cuenta con la posibilidad de salvar los modelos y simulaciones creados, a diferencia de la mayoría de versiones de prueba o evaluación [URL03].

Con todo esto, Vensim sería una buena alternativa a EJS, pero dos pequeños detalles le restan valor para el proyecto. La distribución PLE no permite crear más de una vista para un mismo modelo, y dada la enorme complejidad de algunos de los modelos a simular, es evidente que será necesario el despliegue de múltiples vistas. Más aún considerando los requisitos de vistas desplegadas en modo SVGA y que dispongan de visualizaciones tanto científicas como de carácter más intuitivo. Además, Vensim PLE no ofrece ninguna facilidad o mecanismo para la distribución de las simulaciones creadas, a diferencia de EJS y Stella.

4.2.3. *Stella & Netsim*

Stella es una herramienta de *isee systems* para el modelado de sistemas con finalidades educativas, de forma parecida a EJS. Su sistema de modelado es probablemente más simple aún, al ser principalmente visual, aunque esto puede hacer que algunos modelos lleguen a ser algo “engorrosos”. Ofrece potentes opciones de visualización y creación de interfaces gráficas. *Netsim*, por su parte, es la herramienta de *isee systems* para la exportar los modelos y simulaciones en múltiples formatos y medios, incluidos páginas web e Internet [URL04].

Teniendo en cuenta las prestaciones aparentes de *Stella* y *Netsim*, son una alternativa pareja a EJS. Incluso puede que superior si se tuviesen en cuenta prestaciones que no nos interesan mucho la verdad, como la posibilidad de habilitar contraseñas y otras protecciones en los modelos. Pero esto no compensa en forma alguna el prohibitivo coste que supondría la adquisición de este conjunto de programario, alrededor de ochocientos dólares.

4.3. ¿Como funciona EJS?

EJS permite modelar sistemas y crear las simulaciones pertinentes, y distribuir éstas con la documentación necesaria. Para hacer esto dispone de una serie de herramientas y funciones para crear un modelo basado en un sistema, la GUI de la simulación para entorno WIMP, y la documentación de éste. Finalmente, mediante una

serie de funciones automatizadas, las simulaciones pueden ser empaquetadas para su distribución como ejecutables independientes, como *applets* para páginas HTML, o incluso como páginas web generadas automáticamente.

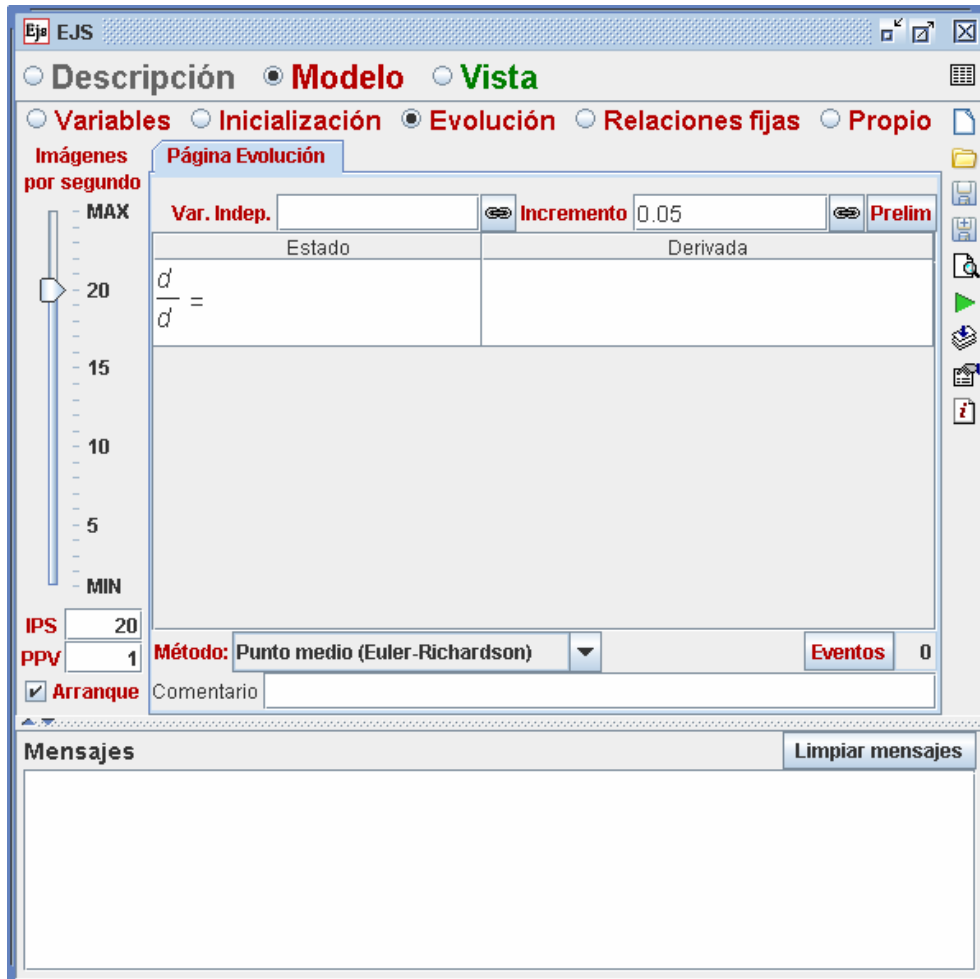


Fig. 4: Interfaz de EJS, creando página de EDO

Un primer vistazo a la interfaz principal de EJS (figura 4) puede ser engañoso, ya que parece excesivamente simple y reducida, impropio de una aplicación potente. La documentación se crea en un pequeño editor de páginas de texto en formato HTML en la pestaña de *Descripción*, donde se pueden incrustar sin problema imágenes y otros objetos. Bajo la pestaña de *Vista* se haya un potente editor de interfaz gráfica, del cual se darán algunos detalles.

La pestaña de *Modelo* despliega todo lo necesario para modelar el sistema y suele contener el grueso del código de la simulación. Las variables se definen en la pestaña de *Variables*, empleando los tipos básicos habituales de *Java*. De ser necesario es enteramente posible definir o emplear cualquier otro tipo usable en *Java* mediante el

tipo *Object*. En la misma pestaña de *Variables* es posible inicializarlas con un valor por defecto, aunque en caso de ser necesaria una inicialización más compleja o especializada, se puede emplear la pestaña *Inicialización* para introducir cualquier cantidad código necesaria. Cualquier número de páginas con código introducido en *Inicialización* será ejecutado de izquierda a derecha, siendo siempre éste el orden usado en cualquier parte de EJS donde que se encuentren múltiples páginas o pestañas de código a un mismo nivel.

En la pestaña de *Evolución* se define el comportamiento del modelo. Se puede introducir cualquier cantidad de páginas de código *Java* o ecuaciones diferenciales ordinarias, abreviadas como EDO –ODE en inglés-. Una de las mejores funcionalidades de EJS es precisamente el editor de EDO, que permite introducir un conjunto de EDO todo lo extenso y complejo que se desee, y escoger entre 8 algoritmos distintos para su resolución. La pestaña de *Relaciones Fijas* permite introducir código *Java*, al igual que las páginas de código de *Evolución*, con una diferencia: las condiciones introducidas aquí se evalúan y mantienen no solo en tiempo de ejecución sino en todo momento, permitiendo que actúen en respuesta a la interacción del usuario con la simulación pausada.

Por último, la pestaña *Propio* permite introducir páginas de código con funciones auxiliares. Esto es especialmente útil para usar funciones especiales dentro de las páginas de EDO, o para simplemente asociar funciones auxiliares a eventos de la interfaz de usuario o facilitar la reusabilidad de código dentro de la simulación.

En cuanto a la propia ejecución de las simulaciones, EJS emplea un esquema bastante simple y bien definido, ayudando aprovechamiento máximo de sus características. Lo que sucede exactamente cuando se ejecuta una simulación es lo siguiente (ver figura 5):

- Se crean las variables con los valores dados por defecto y de haberlo, se ejecuta el código de *Inicialización*.
- Se evalúan las *Relaciones Fijas*, dada la posibilidad de que el valor inicial de alguna variable dependa del valor inicial de otras variables.
- Se crean las vistas de la simulación y se visualizan en pantalla. En este momento se dice que el modelo se encuentra en su estado inicial, reflejado en la vista, y la simulación aguarda hasta que se ejecute un paso de simulación o el usuario interactúe con ella.

- En caso de tener lugar un paso de simulación, se evalúan las ecuaciones y/o código de *Evolución*, e inmediatamente después, las *Relaciones Fijas* de haberlas. Una vez hecho esto, se alcanza un nuevo estado de modelo en un instante de tiempo nuevo, y se actualizan las vistas.
- Si en vez de un paso de simulación lo que tiene lugar es interacción por parte del usuario, se modifica la variable del modelo o ejecuta el código asociado a la acción que realizó. En cualquier caso, se evalúan las *Relaciones Fijas* inmediatamente después, dando lugar a un nuevo estado del modelo en el mismo instante de tiempo. Este nuevo estado es reflejado por la vista de la simulación.

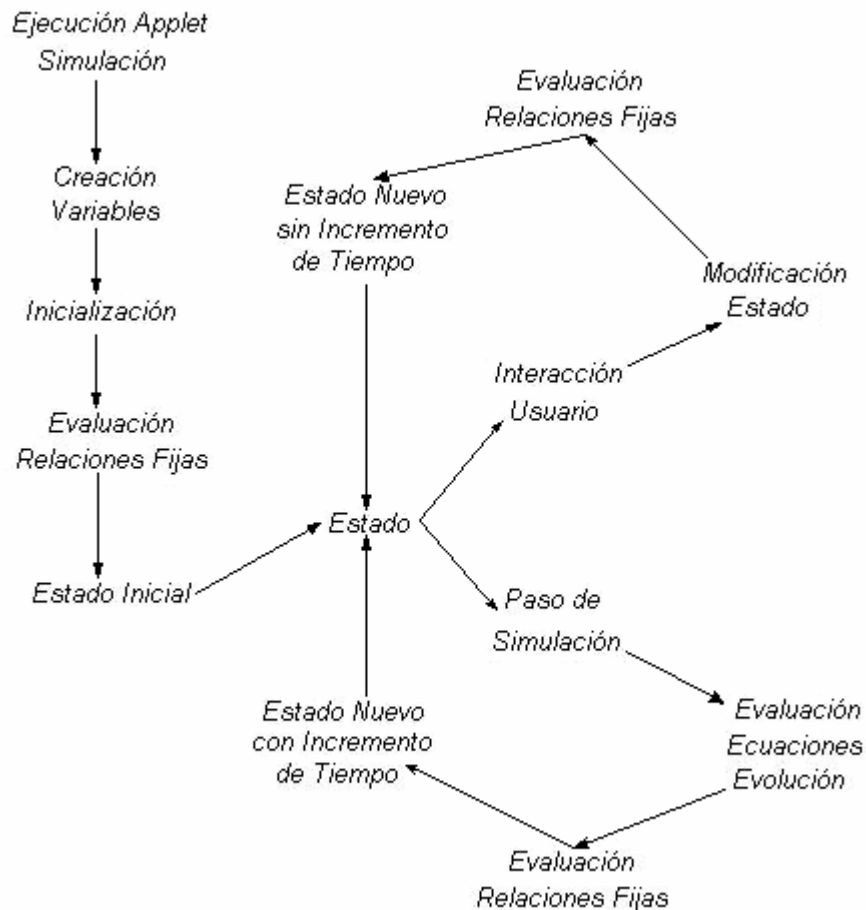


Fig. 5: Flujo de ejecución de una simulación EJS


4.4. Elementos visuales empleados


El editor de vistas de EJS es una funcionalidad realmente completa y que ofrece resultados de gran calidad. Su funcionamiento y gran usabilidad están pensados con


un claro enfoque a programadores no profesionales, resultando bastante intuitivo. De hecho simplemente hay que ir creando un árbol de elementos visuales, siendo imposible colgar hijos de un padre no factible. No obstante, esto no le resta complejidad en la implementación de ciertos tipos de vistas, especialmente en los tridimensionales, los cuales exigen unos conocimientos mínimos previos asociados a este tipo de visualización.


Lejos de detallar las opciones o funcionamiento de este editor, a continuación se listan algunos de los elementos visuales y de control empleados en la implementación de las simulaciones, agrupados según se consideren elementos de control o visualización.


4.4.1. Elementos de control


Botón (*Botón* ): Un simple botón. Permite llamar a ejecución un fragmento de código pequeño, aunque siempre es mejor llamar a funciones. Puede mostrar texto y/o una imagen. Un uso típico es llamar a la función `_play()` entre otras.

Botón múltiple (*BotonDosEstados* ): Como el botón normal, pero con mayores funcionalidades. Se define con la mayoría de atributos por duplicado, y en cada pulsación sobre el, cambian de estado los atributos que se le aplican. El mejor ejemplo es el botón *Play/Pause*.

Casillero Seleccionable (*Selector* ): Una simple casilla de verificación o *checkbox* tradicional. Se pueden definir el aspecto visual hasta con imágenes como los botones, y distintas acciones según el estado como el botón múltiple.

Casillero de Sonido (*SelectorSonido* ): Similar al casillero seleccionable, pero con un añadido. Se le asocia un archivo de sonido (.m3, .wav...), que se reproduce automáticamente si el selector está en estado *Sí*. Por lo demás idéntico al Selector habitual.

Deslizador (*Deslizador* ): asociado a una variable, éste permite modificar el valor dentro de un rango definido. Permite definir también valores iniciales, código asociado a su manipulación y amplias opciones en cuanto a formato gráfico para hacer más agradable su visualización.

Lista seleccionable (*Lista* ): una lista para seleccionar una opción de entre las disponibles. Tanto las opciones posibles como la variable asignada por la selección son tipo *String*.

Campo Numérico Editable (*CampoNumerico*²⁰): el elemento *CampoNumerico* muestra el valor de una variable, de acuerdo a un formato indicado, como cabecera, unidades, precisión si es tipo *Double*... Además de poder definir su aspecto visual con múltiples opciones (tipo de letra, tamaño, color, fondo...) tiene un atributo crítico: *Editable*. Este booleano, de ser cierto, permite que se introduzca un valor en el campo en lugar del que posee la variable, y asignárselo si es un valor válido -desde un punto de vista de tipos *Java*-. Si *Editable* es falso, tan sólo mostrará el valor de la variable.

Pestañas (*PanelConSeparadores*²¹): las típicas pestañas/fichas o paneles. Cada elemento hijo colgado de este padre creará un panel nuevo, por ello, si se desea que una pestaña o panel muestre más de un elemento, será preciso agruparlos en contenedores.

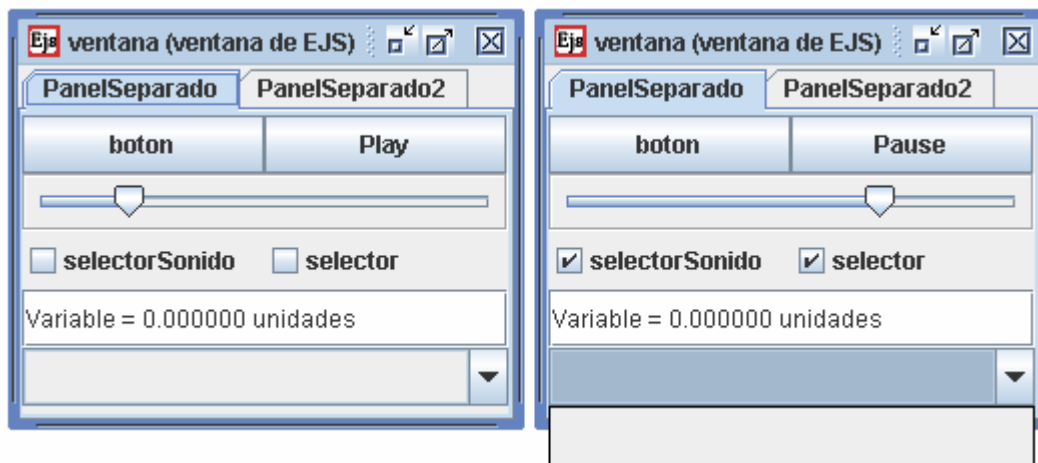




Fig. 6: Ejemplos de elementos de Control de EJS


4.4.2. Elementos de visualización


Ventana (*Ventana*²²): elemento básico de la visualización. Todo *applet* visible tiene al menos una. En el caso de *applets* incrustados en páginas HTML la denominada ventana principal es la incrustada, dejando de ser una ventana independiente. Si la ventana principal se cierra, se termina la simulación.

Ventana de diálogo (*VentanaDialogo*²³): como la ventana normal, con la diferencia de que siempre permanece al frente. No se puede minimizar, pero se puede cerrar, siempre que el atributo *Cerrable* lo permita.

Etiqueta (*Etiqueta* ): permite introducir un cuadro de texto -con el formato gráfico deseado- como uno más de los elementos gráficos de la simulación.

Panel de Dibujo (*PanelDibujo* ): un contenedor para los elementos de dibujo e imágenes en dos dimensiones.

Panel de Gráficas (*PanelConEjes* ): en contra de lo que podría intuirse, permite introducir prácticamente cualquier elemento de dibujo bidimensional. Amplia la funcionalidad de panel de dibujo añadiendo los Ejes y su tratamiento numérico y visual.

Traza (*Traza* ): es una colección o *buffer* de puntos formados a partir de dos “variables” de entrada. En cada paso de simulación, se toman los valores de estas “variables” y se usan para definir un punto nuevo. Se puede limitar el tamaño a un número determinado, a partir de ese máximo, se descartan puntos tomados como en una cola *FIFO*. Dispone de múltiples opciones de visualización, y es el elemento empleado para hacer dibujar las gráficas en los elementos *PanelConEjes*.










Elementos de dibujo varios (*Flecha* , *Partícula* , *Polígono* ): permite crear una flecha, partícula o un polígono respectivamente. Al ser dinámicos, pueden cambiar durante la ejecución, no solo de posición, sino cualquier atributo, siendo posible incluso definir colores dinámicos.















Imagen (*Imagen* ): este contenedor permite insertar imágenes en un panel de dibujo o gráficas. Dispone de opciones básicas de rotación y otras transformaciones, y soporta multitud de formatos de imagen, incluso animaciones (como ficheros .gif).

Conjunto de elementos (*Conjunto* “Elemento” , , , ...): este elemento permite definir una colección de elementos de un tipo. Prácticamente todos los elementos de dibujo bidimensional pueden disponerse como un conjunto, excepto el polígono y otros especiales no usados en este proyecto.

Panel en 3D (*PanelDibujo3D* ): el elemento contenedor para visualizaciones tridimensionales, inclusive gráficas y trazas en tres dimensiones, ya que no hay panel de ejes en 3D. Requiere de cierta pericia y conocimiento de visionado en 3D ya que es preciso definir el espacio a visualizar y la cámara: tanto punto de vista, foco, distancias... Cuenta además con un inconveniente bastante molesto, sobretodo a la

hora de realizar gráficas tridimensionales: a diferencia de otros paneles de dibujo, no dispone de autoescalado. Esto hace preciso el conocer el espacio máximo que empleará la visualización que deseamos, y que sea el usuario quien deba disponer o el suficiente espacio de visualización, o los mecanismos de escalado.

En cuanto a dibujo en 3D, existen versiones específicas de todos los elementos 2D citados -y la mayoría de los no usados en el proyecto- para ser empleados en 3D, inclusive de los conjuntos. De hecho, se dispone adicionalmente de elementos para las siguientes figuras tridimensionales, así como la formación de conjuntos de éstas:

- *Plano3D* , *ConjuntoPlanos3D* 
- *Superficie3D* , *ConjuntoSuperficies3D* 
- *Caja3D* , *ConjuntoCajas3D* 
- *Esfera3D* , *ConjuntoEsferas3D* 
- *Elipsoide3D* , *ConjuntoElipsoides3D* 
- *Cono3D* , *ConjuntoConos3D* 
- *Cilindro3D* , *ConjuntoCilindros3D* 

5. Modelado para predicción de Nivel Sónico Equivalente, L_{eq}

5.1. Introducción

Desde hace ya unos años, un tema del que las múltiples administraciones están tomando conciencia es el de la contaminación acústica. Este tipo de contaminación, hasta hace no mucho ignorado, se está convirtiendo en un problema en las grandes urbes, tanto por su magnitud, como por sus consecuencias. Insomnio, stress, problemas psicológicos y fisiológicos... que se multiplican cada vez a un ritmo mayor debido a una gran proliferación de contaminantes acústicos dentro de núcleos urbanos.

El modelo propuesto permite predecir y/o evaluar el impacto acústico de una vía de gran volumen de tráfico, no solo de carácter interurbano, como una autopista, sino también vías dentro de poblado como rondas de circunvalación. Esta predicción será en base a la distancia que separa el receptor de esta vía, así como estadísticas completas sobre los tipos de tráfico rodado en cada hora del día. De igual forma, empleando los mismos datos estadísticos, es posible visualizar la densidad de tráfico ponderado. Esta medida resulta de utilidad para estudiar como al variar la distribución de la densidad de tráfico, un mismo volumen diario de tráfico rodado puede producir diferentes niveles de contaminación acústica y malestar al ser humano.

La simulación trabajará con datos estadísticos del tráfico para cada hora del día, a saber: velocidad media del tráfico, volumen de vehículos de hasta 2 ejes - generalizados como automóviles- y volumen de vehículos a partir de 3 ejes - generalizados como camiones-. Por ello será necesario que cuente con mecanismos para la adquisición de estos datos, así como de la distancia que separará el receptor de la vía. Visualmente, la simulación debe poder mostrar sus resultados tanto de una forma concisa y precisa por un lado, como de una forma más clara e intuitiva por otro.

5.2. Descripción del modelo matemático

Este modelo se basa principalmente en el estudio de la “energía equivalente al nivel de sonido”, también L_{eq} a partir de ahora, y la estimación de la “densidad media de

tráfico ponderado”, $Av_{traffic}$. Estas constituyen las principales variables de estado y salidas de la simulación:

- **Energía equivalente al nivel de sonido (L_{eq})**
- **Densidad media de tráfico ponderado ($Av_{traffic}$)**

La *energía equivalente al nivel de sonido* es una medida de la molestia acumulada debido al ruido durante una hora, en este caso, debido a la contaminación acústica del tráfico rodado. Se expresa en [dBA], una variación de la unidad [dB] usada en acústica, la cual es referenciada a un potencial [dB 0] fijado en el umbral de audición humana medio; y filtrada a través de filtros A para tomar principalmente las frecuencias más dañinas para el oído humano. Así pues, L_{eq} expresa esencialmente un nivel de molestia potencialmente dañino para el oído humano desprotegido.

La *densidad media de tráfico ponderado*, $Av_{traffic}$ es una medida de la relación entre el tráfico ponderado a razón de ejes y la velocidad media, permitiendo el estudio de la correlación entre el tráfico (como volumen ponderado de vehículos), la velocidad media, y el impacto sobre L_{eq} .

El modelo estudiado [DAV07], se basa en ecuaciones empíricas desarrolladas para predecir como el tráfico puede perturbar la vida humana y constituir un riesgo para los seres humanos que desarrollan parte de su actividad cerca de vías rodadas no protegidas acústicamente. Para predecir la *energía equivalente al nivel de sonido*, es necesaria la mínima distancia desde la vía hasta el “punto de recepción”, así como una estimación de los volúmenes de tráfico para automóviles -vehículos motorizados hasta dos ejes- y camiones -vehículos motorizados de 3 o más ejes-, así como la velocidad media en cada hora a estudiar.

La introducción de la *densidad media de tráfico ponderado* en el modelo fue realizada con el fin de poder evaluar mejor los resultados obtenidos del cálculo de L_{eq} . Por ello, con el fin de compatibilizar las magnitudes de densidad de tráfico obtenidas con el resto del modelo L_{eq} y sus ecuaciones, el volumen de automóviles y camiones fue ponderado análogamente a la expresión dada de L_{eq} .

Así, el modelo se describe mediante las siguientes ecuaciones y variables:

$$L_{eq} = 42.3 + 10.2 \log(V_c + 6V_t) - 13.9 \log D + 0.13S$$

$$Av_{traffic} = \frac{(V_c + 6V_t)}{S}$$

where

L_{eq} = energy equivalent sound level during one hour, dBA

$Av_{traffic}$ = average wheel traffic flow during one hour

V_c = volume of automobiles (four tires only), vehicles/hour

V_t = volume of trucks (six or more tires), vehicles/hour

D = distance from edge of pavement to receiver, m

S = average speed of traffic flow during one hour, km/h

Con un diagrama de bloques describiendo su comportamiento de la forma siguiente (figura 7):

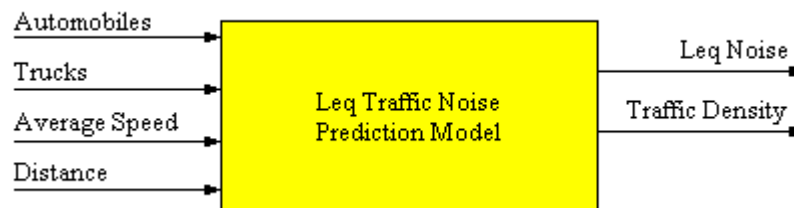


Fig. 7: Diagrama de bloques del modelo

Esto nos da como variables de salida:

- L_{eq} = energía equivalente al nivel de sonido = variable [dBA]
- $Av_{traffic}$ = densidad media de tráfico ponderado = variable [ratio]

Cuyos valores dependen de:

- volumen de tráfico para cada hora, tanto para automóviles (av_{hour} = [vehículos/hora]) como para camiones (tv_{hour} = [vehículos/hora])
- velocidad media del tráfico (as_{hour} = [km/h]) en la vía estudiada, para cada hora
- mínima distancia desde la vía hasta el punto de recepción humana ($distance$ = [m])

5.3. Aspectos técnicos

Dado que éste fue el primer modelo implementado bajo EJS para el proyecto, había ciertos aspectos del entorno de trabajo no asimilados aún. Siguiendo criterios habituales en el diseño de programas, todas las variables con valores para cada hora se dispusieron en vectores, con el fin de permitir un diseño de código más limpio y menos redundante.

No obstante, la posterior implementación de las vistas reveló la imposibilidad de acceder a una variable para editarla direccionándola por una expresión en el uso de elementos *CampoNumerico*. Es decir, un cuadro como el de la figura 8 permitiría acceder al valor *av[0]*, pero no editarlo.

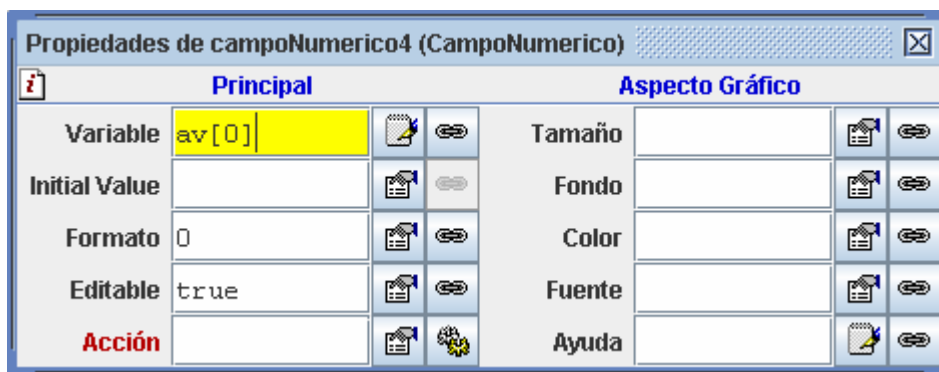


Fig. 8: Ejemplo de edición de propiedades de un *campoNumerico*

Esto obligó a implementar un juego adicional de variables, concretamente una por cada valor necesario para cada hora, para poder mantener la vista *Traffic Stats* con posibilidad de edición y mostrar todos los valores a la vez. Pese a inducir un factor de redundancia del doble en este caso fue una solución satisfactoria, dado el reducido número de variables a duplicar. El rendimiento de la simulación tampoco se ha visto perjudicado, dado que tampoco se llega a marcar un valor de máxima valor de ejecución posible -a ritmo de ejecución *MAX* se tendrían al menos 25 iteraciones por segundo, dándose un ciclo de simulación de un día por segundo y dificultando la percepción de muchos de los cambios de la representación pictórica, ya que el ojo humano no alcanzaría a ver la representación del estado de una hora antes de que cambiase al de la siguiente-.

5.4. Visualización y control de la simulación

Uno de los puntos ya mencionados y que más se ha tenido en cuenta a la hora de implementar la simulación es la compatibilidad con navegadores y sistemas en baja resolución. Con el fin de permitir esta compatibilidad con resoluciones bajas -800x600 puntos por ejemplo- se ha procurado mantener un área de *Applet* reducida.

Esta limitación ha sido sorteada para poder implementar un sistema de control y visualización completo mediante la introducción de “subvistas”, o pequeñas ventanas adicionales. De este modo se puede manejar una “área útil” grande, manteniendo un área en pantalla pequeño.

Así pues, tenemos las siguientes tres vistas asociadas a la simulación:

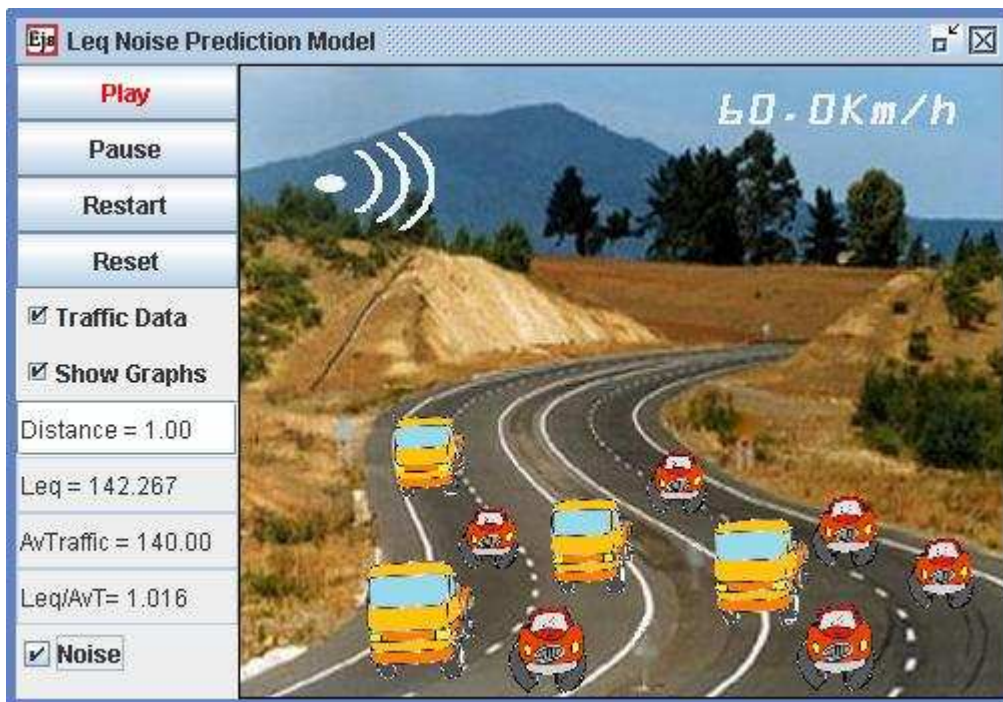


Fig. 9: Vista principal del modelo

La vista principal dispone de la mayoría de elementos de control, así como ofrece toda la información de salida instantánea. Mediante un panel de dibujo, se muestra una representación pictórica del tráfico y su evolución. Asimismo se puede observar la velocidad media instantánea y la evolución de L_{eq} en el pictograma de ondas de sonido. Los botones permiten el control del flujo, así como el reiniciado de la simulación y el “reseteado” -botón *Reset*-, deshaciendo los cambios editados en los datos sobre el tráfico. Los casilleros seleccionables permiten activar la visualización de las otras vistas o del sonido de tráfico ambiental.

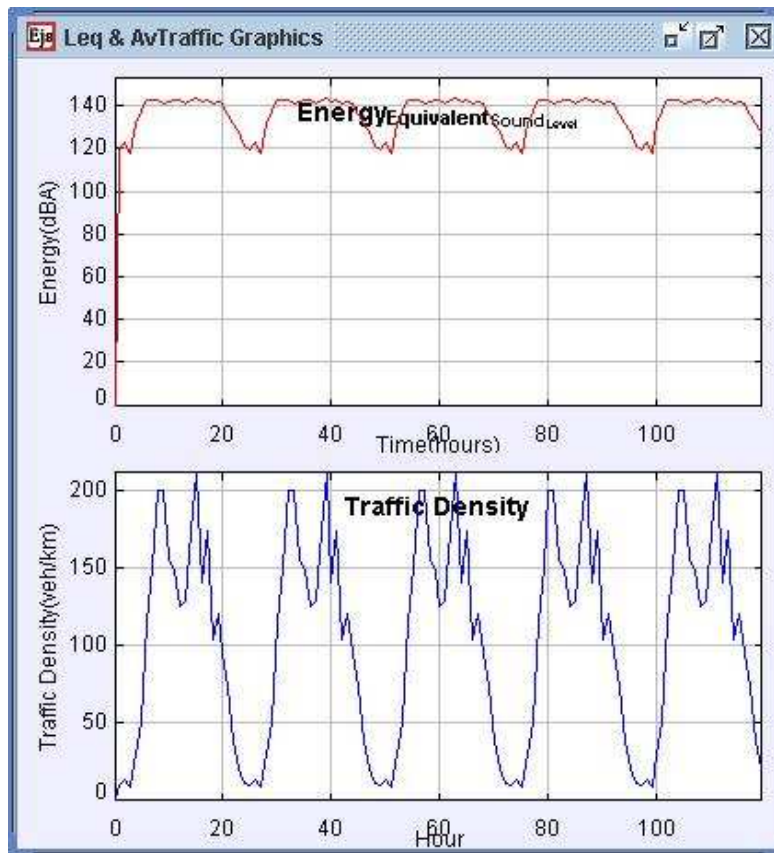


Fig. 10: Vista con gráficas de evolución de L_{eq} y $AV_{Traffic}$

La segunda vista, figura 10, muestra simplemente las gráficas correspondientes a la evolución de L_{eq} y $AV_{Traffic}$ en función del tiempo -en horas-.

La última vista, *Traffic Data Display* (figura 11) contiene un conjunto de visualizadores numéricos editables -elemento *CampoNumerico* de EJS-. Estos componentes - también empleados en la vista principal de la simulación- permiten mostrar el valor de una variable, así como editar su valor. De este modo, se emplean tanto para mostrar los valores medios empleados en la simulación por defecto, como para permitir al usuario introducir sus propios datos para hacer su propia simulación.



	AutoVolume	TruckVolume	AvSpeed
0h	500	50	80
1h	450	45	80
2h	400	100	80
3h	300	50	80
4h	500	300	80
5h	1000	500	80
6h	2000	1000	70
7h	3000	1000	60
8h	4000	1000	50
9h	4500	750	45
10h	3750	800	55
11h	3500	900	60
12h	3000	850	65
13h	3500	700	60
14h	3500	1000	55
15h	4000	1100	50
16h	3000	900	60
17h	3500	1000	55
18h	2500	700	65
19h	3000	800	65
20h	3000	600	70
21h	2000	400	60
22h	1700	150	70
23h	1000	100	80

Fig. 11: Estadísticas horarias del tráfico

6. Modelado de un secador solar de sultaninas

6.1. Introducción

El secado de productos agrícolas ha sido, desde tiempos remotos, de suma importancia para la preservación de los alimentos para los seres humanos. En un mundo de plagas inmunes a los pesticidas convencionales debido a su uso abusivo, catástrofes naturales meteorológicas inducidas por el cambio climático, y procesos de desertización imparables, la producción de alimentos dista de ser estable o garantizada. Esto da mayor importancia aún a la capacidad de preservar alimentos de forma efectiva e higiénica, como medio de combatir la inseguridad en el suministro de alimentos a escala mundial.

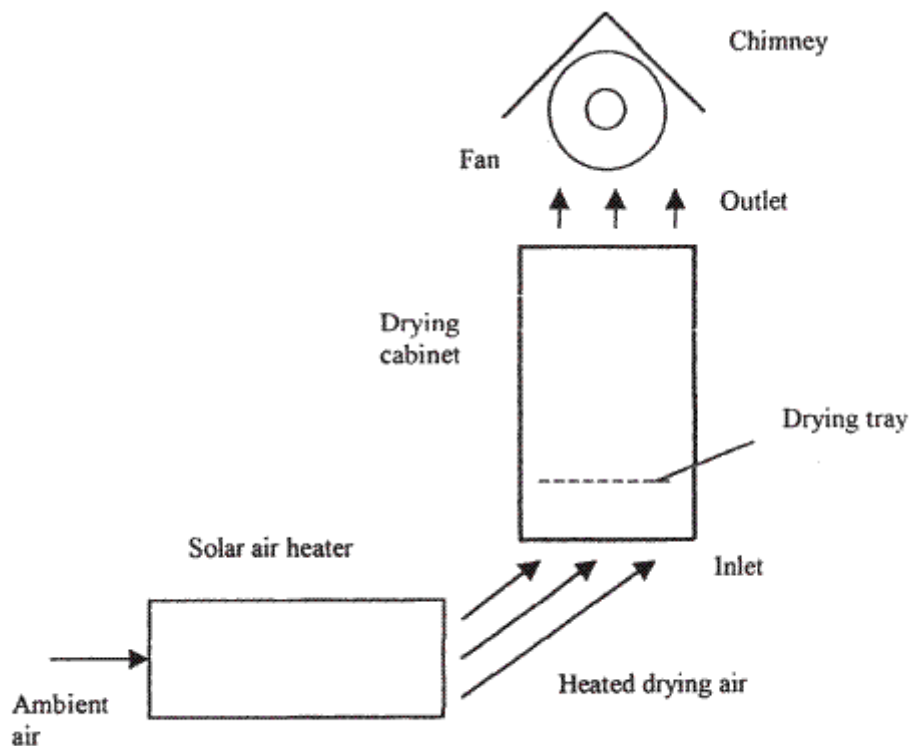
De entre los distintos métodos, el secado solar ha sido y aún es el más común en los países tropicales y subtropicales. No obstante, sin la protección adecuada contra lluvias, agentes contaminantes atmosféricos y polvo, plagas y otros riesgos, la calidad de los alimentos se puede degradar hasta ser no aptos para el consumo. Las consecuencias de esta pérdida de calidad no solo son de carácter económico, sino que puede llegar a producir problemas de abastecimiento alimentario en zonas críticas.

Por todo esto resulta de gran importancia el estudio de la eficiencia y eficacia de los distintos métodos de preservación de alimentos. Está simulación concretamente estudia el modelo matemático de un secador solar de sultaninas, con cabina para aislar el producto del entorno durante el proceso, protegiéndolo de los riesgos antes mencionados. Este dispositivo cuenta además con un ventilador que produce un flujo de aire de velocidad regulable. El estudio de este flujo de aire y su velocidad, permite predecir la temperatura del secador solar, y estimar la humedad aún presente en las sultaninas. El análisis de estos datos permite evaluar la eficiencia y eficacia del dispositivo secador en función de la velocidad de aire inducida.

6.2. Descripción del modelo matemático

Este modelo [YAL01] matemático estudia el proceso de secado solar de sultaninas empleando un secador solar de convección indirecta forzada, en las condiciones habituales de Antalya, en Turquía.

Un secador solar de convección indirecta forzada –esquema en figura 12- consiste en un calentador de aire y una cabina de secado. El calentador de aire usado en los experimentos modelados es un cuadro de madera aislada, con una lámina de hierro pintada de negro como panel de absorción para absorber la radiación solar incipiente; y una lámina de plástico transparente como cubierta para evitar pérdidas de calor. La cabina de secado, también hecha de madera aislada, dispone en el fondo de bandejas de rejilla metálica para contener las sultaninas, y de un ventilador eléctrico en una campana de extracción. De este modo, el aire ambiental entra en el secador debido a la corriente producida por el ventilador. Este aire que entra pasa inicialmente por el calentador de aire, calentándose antes de entrar en la cabina de secado.



Schematic diagram of a solar cabinet dryer

Fig. 12: Esquema de un secador solar de sultaninas

El modelo estudia la eficacia del secador solar a distintas velocidades de flujo de aire, determinadas por la velocidad del ventilador. La variación en la velocidad del aire determina el comportamiento y evolución de las variables del modelo a lo largo del tiempo, a saber:

1. Temperatura dentro de la cabina de secado (T)

2. Relación de humedad (MR)

La temperatura dentro de la cabina de secado se simula mediante una variable de estado, como también la relación de humedad. La relación de humedad, o también *MR* a partir de ahora, refleja aproximadamente el porcentaje de la humedad inicial aún presente en las sultaninas. Todas las ecuaciones del modelo fueron derivadas del estudio de múltiples análisis de regresión, de los cuáles se escogieron solo aquellos cuyas curvas de evolución tomaran una mayor correlación con los datos empíricos. Estos datos fueron obtenidos mediante una serie de experimentos llevados a cabo en Antalya (36°53'N,30°42'E), Turquía.

Los experimentos tuvieron lugar durante los periodos de Junio a Agosto de los años 1997 y 1998. Un total de 22 experimentos fueron realizados, agrupados en tres conjuntos, en función de la velocidad del flujo de aire inducido por el ventilador. Así hicieron múltiples experimentos a velocidades de 0.5 m/s, 1 m/s y 1.5 m/s, en los que se distribuía una capa uniforme de 16kg de uva/ m² y se estudiaba su relación de humedad al final de cada día. Se registraron durante tales experimentos temperaturas ambientales de entre 32.15°C y 33.32°C, humedades relativas del 66.43 hasta el 75.7%, y una radiación solar variante entre 790.3 y 802.0W/m². Para calcular la relación de humedad, el contenido húmedo fue registrado como el porcentaje húmedo base y luego convertido a kg de agua por kg de materia seca para el modelado.

La velocidad del aire ($V=[m/s]$) y la temperatura ($T=[°C]$) están relacionadas por una función cóncava, que muestra como la temperatura dentro de la cabina de secado varía a medida que aumenta la velocidad del aire. Esta función se describe en el modelo como una ecuación polinómica de segundo orden, obtenida por análisis de regresión cuadrática de los datos empíricos tomados durante los experimentos (tabla 1).

Model	Name of model
$MR=\exp(-kt)$	Newton
$MR=\exp(-kt^n)$	Page
$MR=\exp[-(kt)^n]$	Modified Page
$MR=a \exp(-kt)$	Henderson and Pabis
$MR=a \exp(-kt)+c$	Logarithmic
$MR=a \exp(-k_0t)+b \exp(-k_1t)$	Two term
$MR=a \exp(-kt)+(1-a)\exp(-kat)$	Two term exponential
$MR=1+at+bt^2$	Wang and Singh

Tabla 1: modelos de regresión empleados para estimar la ecuación de MR

La relación de humedad se calcula con un modelo de ecuación de dos términos, obtenido de entre el estudio de ocho modelos de regresión diferentes. El modelo seleccionado mostró el mejor coeficiente de correlación de entre todos, y sus χ -cuadrados reducidos determinaron que se ajustaba adecuadamente a las curvas de datos, demostrando ser la opción óptima par la simulación (tabla 2).

	r	χ^2
Newton model $k=0.042707$	0.966	6.30×10^{-3}
Page model $k=0.088790 \ n=0.785434$	0.974	4.96×10^{-3}
Modified Page model $k=0.000175 \ n=0.000008$	0.673	5.20×10^{-2}
Henderson and Pabis model $a=0.941475 \ k=0.042707$	0.969	5.82×10^{-3}
Logarithmic model $a=0.915547 \ k=0.042707 \ c=0.040518$	0.970	5.68×10^{-2}
Two-term model $a=0.194286 \ k_0=6.351304 \ b=0.805715 \ k_1=0.033952$	0.979	4.55×10^{-3}
Two-term exponential model $a=0.159830 \ k=0.226307$	0.973	5.00×10^{-3}
Wang and Singh model $a=-0.029602 \ b=0.000231$	0.944	1.02×10^{-2}

Tabla 2: Resultados de los análisis de regresión para buscar MR

Durante los cálculos, MR fue simplificado a M/M_0 (masa húmeda/masa húmeda inicial), en lugar de $(M-M_e)/(M_0-M_e)$, ya que la humedad relativa del “aire de secado” fluctuaba continuamente durante el proceso de secado. El modelo muestra una relación de humedad decreciente en todo momento, así como la velocidad de secado, lo cual es coherente con el hecho de que al haber menos humedad, es más difícil su eliminación.

El modelo obtenido se describe matemáticamente mediante las siguientes ecuaciones:

$$MR = \frac{M}{M_0} a \exp(-k_0 t) + b \exp(-k_1 t)$$

$$T = -15.8V^2 + 25.1V + 30.3$$

where :

$$a = 0.336 - 0.004T$$

$$k_0 = 7.703 - 8.717 \ln(V)$$

$$b = 0.806V^{-0.039}$$

$$k_1 = -0.141 + 0.048 \ln(T)$$

Y su comportamiento es descrito siguiente diagrama de bloques:

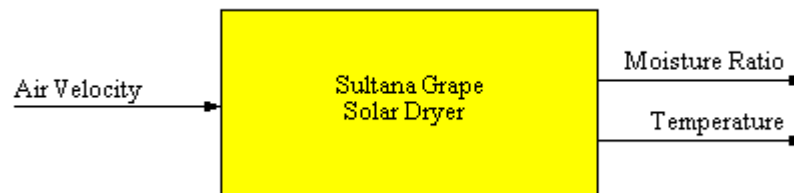


Fig. 13: Diagrama de bloques para el modelo del secador solar de sultaninas

Donde las variables de salida son:

- T = temperatura dentro de la cabina de secado = variable [°C]
- MR = relación de humedad de las sultaninas secándose = variable [ratio]

Cuya evolución depende de la variable de entrada

- V = velocidad del aire = variable [m/s]

6.3. Visualización y control

La vista principal (figura 14) y despliegue de elementos de control para la simulación de este modelo son bastante simples y concisos. A nivel gráfico se puede observar una representación pictórica, generada mediante imágenes introducidas y gráficos producidos por EJS, una gráfica de evolución de MR , y los diversos elementos de control y visualización del flujo de datos.

La representación pictórica muestra de forma intuitiva como evoluciona la relación de humedad, apreciándose un descenso en la cantidad de agua de la imagen. Asimismo,

también se puede observar como varía la velocidad de aire introducido en el secador, y como ésta afecta a la temperatura, y su medición en el “termómetro virtual”.

La gráfica a la derecha de la vista muestra la evolución de MR con el fluir del tiempo, expresado en horas. Su evolución finaliza tras un periodo de 90 horas, el cual se considera el límite de aplicabilidad del modelo. Esto es así porque los datos empíricos empleados para el análisis de regresión mediante el cual se obtuvieron los datos solo cubren periodos ligeramente inferiores a las 90 horas.

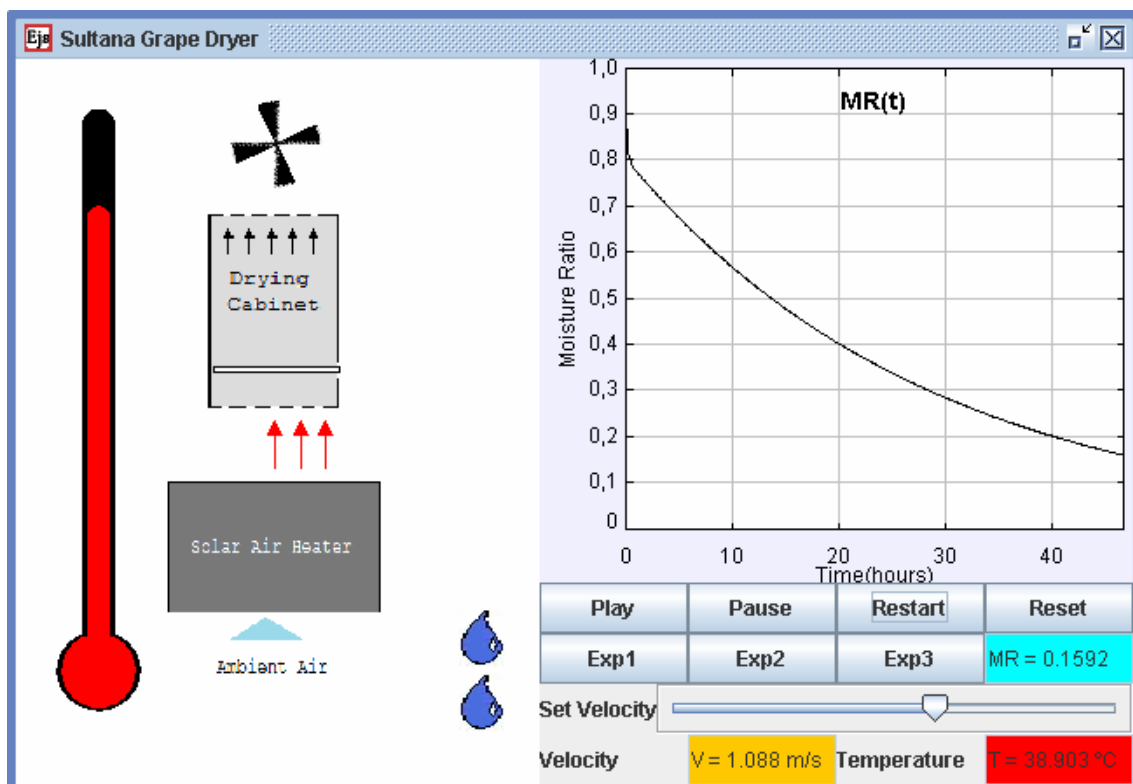


Fig. 14: Vista principal de la simulación de secador de sultaninas

Debajo de la gráfica, se disponen los elementos de control de ejecución habituales, así como la opción de reiniciar la simulación conservando el valor introducido de velocidad del aire -botón *Restart*. Los botones *Exp_i* permiten simular el modelo directamente de acuerdo a los valores de los tres grupos de experimentos que se realizaron, disponiendo la velocidad del aire adecuadamente. Por último, se completa la visualización con los respectivos indicadores, diferenciados por color, de los valores de MR , velocidad de aire y temperatura.

7. Simulador ecológico PZNPO

7.1. Introducción

No siempre son necesarias grandes actuaciones para resolver grandes problemas. Muchas veces sería mucho más efectivo aplicar la simple suma de pequeñas actuaciones y medidas, especialmente en ámbitos demasiado grandes y generales. Este principio, aplicado en la resolución de problemas tanto en ingeniería como en otras disciplinas, también podemos aplicarlo a la hora de estudiar como crear y mantener entornos más sostenibles en términos ambientales.

Con este fin, estudiar lo pequeño y concreto, para obtener información y principios de comprensión de lo grande, se realizó en la UPC de Castelldefels un estudio sobre su estanque. A partir de este estudio y otros modelos sobre dinámicas producidas en estanques y lagos, se modeló el comportamiento de los nutrientes y el plancton en el estanque, dando lugar al modelo PZNPO del estanque de Castelldefels.

Con el fin de aumentar el conocimiento producido y obtener una mayor comprensión de las implicaciones de las variaciones ambientales en el estanque, al modelo PZNPO original se le ha añadido una serie de ecuaciones para poder estimar las emisiones de metano producidas por el estanque. Además, se ofrece la posibilidad tanto de conectar el modelo a una base de datos real creada con información obtenida experimentalmente, o en caso de no estar disponible, la simulación mediante funciones del proceso de obtención de datos.

7.2. Descripción del modelo matemático

El modelo implementado estudia el comportamiento de distintas variables del estanque del Campus de Castelldefels de acuerdo con el modelo ecológico PZNPO propuesto en otros trabajos anteriores, [SOL05]. Específicamente se estudia el comportamiento de:

1. **Fitoplancton (*Phy*)**
2. **Zooplancton (*Zoo*)**
3. **Nutrientes (*N*)**
4. **Fosfatos (*PO*)**

5. Carbono (C)

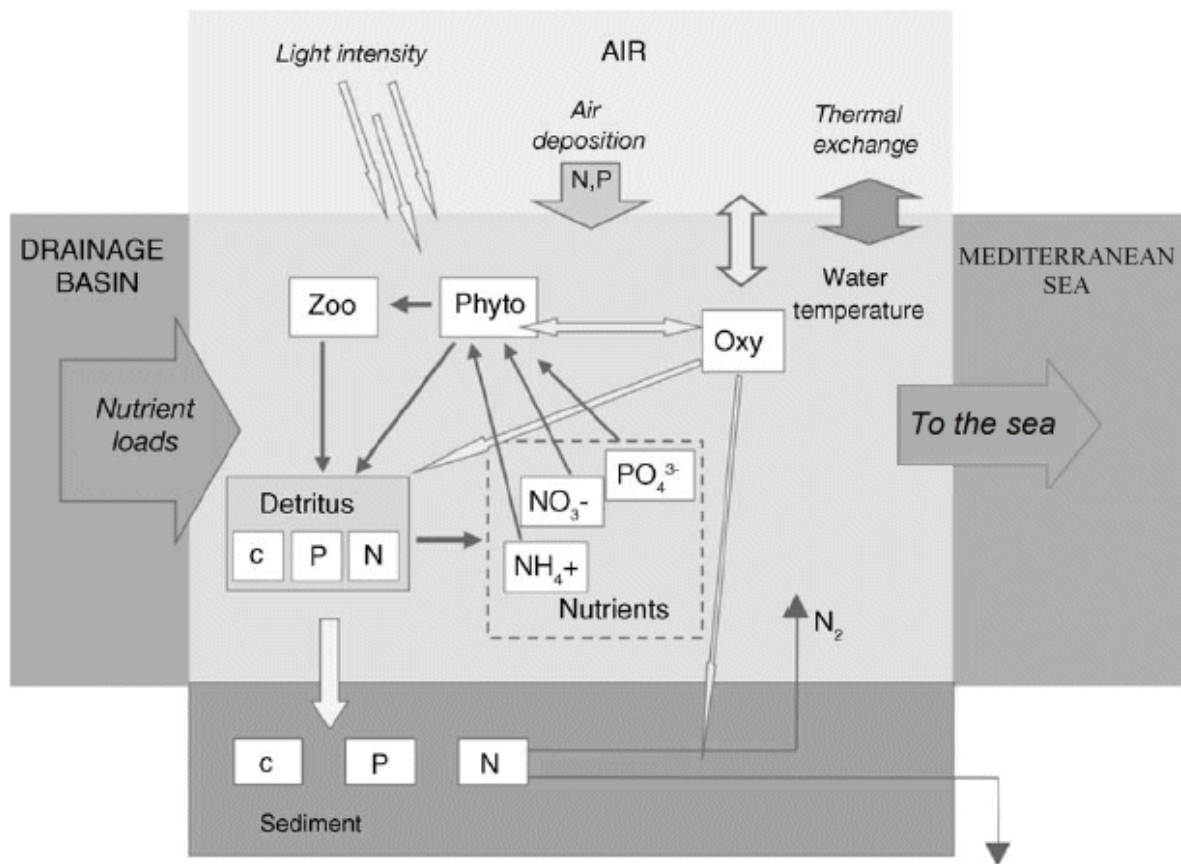


Fig. 15: Esquema de flujos de materia y aguas en el estanque estudiado

A través de éste, es posible estudiar la evolución de los productores primarios (P , Phy), herbívoros (Z , Zoo), los nutrientes más relevantes del entorno (N = nitratos y amonio), fosfatos (PO) como reactivo limitante de la eutrofización en el estanque, carbono (C), y finalmente, el metano producido (CH_4).

El módulo bioquímico describe las principales interacciones entre los cinco compartimentos: agua, fitoplancton, zooplancton, detritus y sedimentos superficiales. Tal y como se aprecia en la figura 15, la cual muestra las variables de estado del módulo y los flujos, los productores primarios se representan mediante una única variable de estado, fitoplancton. El comportamiento de esta variable está determinado por la temperatura del agua, la intensidad lumínica, y las concentraciones de nitrógeno inorgánico disuelto (DIN) y fósforo inorgánico disuelto (DIP), y es afectado por los procesos de depredación e ingesta. La composición elemental del plancton es considerada constante, con las relaciones $C:P$ y $C:N$ iguales a la relación de Redfield. La interacción entre el crecimiento del fitoplancton y el consumo de nutrientes se describe mediante una ecuación cinética de un paso.

El zooplancton también es descrito mediante una única variable de estado, siendo representativa del conjunto de especies que depredan sobre el fitoplancton. La depredación e ingesta que realizan se describe mediante una relación hiperbólica, mientras que la mortalidad, la exudación y respiración de tanto el zooplancton como el fitoplancton se describe como procesos de primer orden: las constantes cinéticas aumentan exponencialmente con la temperatura del agua, de acuerdo con la función Q_{10} .

El compartimento de detritos se describe mediante 3 variables de estado, para contabilizar separadamente las cantidades de C , N y P . Los detritos se hunden, entrando en la superficie de la capa sedimentaria, la cual también es descrita mediante tres variables de estado. El proceso de remineralización de la materia orgánica, que toma lugar tanto en la columna de agua como en la superficie sedimentaria, depende de la temperatura y la disponibilidad del oxígeno disuelto. Este proceso implica el consumo de oxígeno disuelto y conlleva la liberación de nutrientes inorgánicos en la columna de agua. La desnitrificación no se ha tenido en cuenta por simplicidad. Finalmente, las emisiones de metano del sistema se evalúan en función de los nutrientes sedimentarios y los fosfatos, de acuerdo con la ecuación descrita en [WU07].

Matemáticamente el modelo es denotado por un conjunto de ecuaciones diferenciales ordinarias acopladas más una ecuación relacionada con las emisiones de metano:

$$\frac{dPhy}{dt} = growth_{Phy} - resp_{Phy} - egestion - mort_{Phy} - sink_{Phy}$$

$$\frac{dZoo}{dt} = (1 - \gamma)egestion - mort_{Zoo} - excret_{Zoo}$$

$$\frac{dN}{dt} = \gamma egestion + mort_{Zoo} + mort_{Phy} - growth_{Phy}$$

$$\frac{dPO_4}{dt} = -r_{pc} growth_{Phy} + r_{pc} \{ resp_{Phy} + excret_{Zoo} \} + decay_{DetP} + decay_{SedP}$$

$$\frac{dDetC}{dt} = \gamma \cdot egestion + mort_{Zoo} + mort_{Phy} - decay_{DetC} - sink_{DetC}$$

$$\frac{dDetN}{dt} = r_{nc} \cdot (\gamma \cdot egestion + mort_{Phy} + mort_{Zoo}) - decay_{DetN} - sink_{DetN}$$

$$\frac{dDetP}{dt} = r_{pc} \{ grazing + mort_{Phy} + mort_{Zoo} \} - decay_{DetP} - sink_{Phy}$$

$$\frac{dSedC}{dt} = sink_{DetC} + sink_{Phy} - decay_{SedC}$$

$$\frac{dSedN}{dt} = sink_{DetN} + r_{nc} \cdot sink_{Phy} - decay_{SedN}$$

$$\frac{dSedP}{dt} = sink_{DetP} + r_{pc} \cdot sink_{Phy} - decay_{SedP}$$

$$MethaneEmissions = SedC \cdot \alpha + SedN \cdot \beta$$

$$\alpha = 5.073962; \beta = 2.871245$$

where:

$$growth_{Phy} = V_m f(T) \frac{N}{k_N + N} Phy, \quad N = NO_3^- + NH_4^+ \quad \text{phytoplankton growing}$$

$$egestion = R_m (1 - e^{-\Lambda Phy}) Zoo \quad \text{ingestion (zooplankton eats phytoplankton)}$$

$$mort_{Phy} = mPhy \quad \text{phytoplankton mortality}$$

$$mort_{Zoo} = gZoo \quad \text{zooplankton mortality}$$

$$decay_{DetX} = K_{dec_{Det}} \cdot Q_{10} \cdot F_{oxy} \cdot [DetX] \quad \text{detritus decay}$$

$$decay_{SedX} = K_{dec_{Sed}} \cdot Q_{10} \cdot F_{oxy} \cdot [SedX] \quad \text{sediment decay}$$

$$sink_{DetX} = K_{sink_{Det}} \cdot [detX] \quad \text{detritus sedimentation}$$

$$sink_{Phy} = K_{sink_{Phy}} Phy \quad \text{phytoplankton sedimentation}$$

$$f(T) = \log 2 (0.851)(1.066^T), \quad \text{from Eppley (1972)}$$

$$Q_{10} = 1.07^{(T-20)} \quad \text{water temperature influency}$$

$$F_{oxy} = \frac{Oxy}{k_{od} + Oxy}$$

Sirviéndose de los siguientes parámetros:

- Coeficiente de mortalidad del zooplancton, $g = 0.2$ [día⁻¹]
- Constante de media saturación de nutrientes, $k_N = 0.2$ [mmol.m⁻³]
- Constante de media saturación de fosfatos, $k_P = 0.15$ [mmol.m⁻³]
- Constante de media saturación de oxígeno en nitrificación,
 $k_{od} = 62.5$ [mmol.m⁻³]
- Coeficiente de mortalidad del fitoplancton, $m = 0.1$ [día⁻¹]
- Índice de ingesta máxima herbívora, $R_m = 1.0$ [día⁻¹]
- Índice de crecimiento máximo herbívoro, $V_m = 2.0$ [día⁻¹]
- Eficiencia de la ingesta del zooplancton, $\gamma = 0.3$
- Constante de depredación e ingesta herbívora, $\lambda = 0.75$ [m³.mmol⁻¹]
- Razón P/C del fitoplancton (*Redfield ratio*), $r_{pc} = 0.023$
- Razón N/C del fitoplancton (*Redfield ratio*), $r_{nc} = 0.15$
- Velocidad de decaimiento de los detritos orgánicos a 20°C, $K_{decDet} = 0.0048$
- Velocidad de sedimentación del fitoplancton, $K_{sinkPhy} = 0.0016$ [h⁻¹]

Con esto, podemos describir el comportamiento del modelo mediante el siguiente diagrama de bloques:

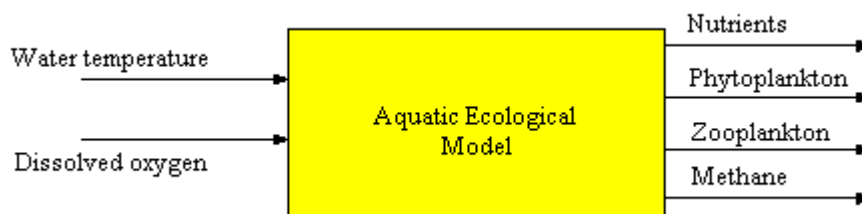


Fig. 16: Diagrama de bloques del modelo del estanque

Donde las variables de salida son:

- N = total de nutrientes disponibles en el sistema = variable [mmol.m⁻³]
- Phy = fitoplancton en el sistema = variable [mmol.m⁻³]
- Zoo = zooplancton en el sistema = variable [mmol.m⁻³]
- PO_{4+} = fosfatos en el sistema = variable [mmol.m⁻³]
- CH_4 = metano liberado por el sistema = variable [mmol.m⁻³]

Las cuales dependen de la temperatura del agua ($T = [^{\circ}C]$) y el oxígeno disuelto en el sistema ($Oxy = [mmol.m^{-3}]$).

7.3. Visualización y control

Dada la ingente cantidad de variables, parámetros y elementos de control requeridos para una adecuada ejecución y visualización de la simulación, se ha procedido implementando una serie de vistas adicionales además de la principal. Estas vistas adicionales se activan mediante casilleros seleccionables, permaneciendo no visibles por defecto. Así tenemos una vista principal, una vista de esquema animado, y finalmente, una serie de subvistas en los cuales se muestran individualmente la evolución de ciertas variables.

La vista principal (figura 17) se divide en varias zona, hallándose a la izquierda de ésta la columna con los controles de ejecución de la simulación. Estos controles incluyen los habituales botones de inicio, pausa y reinicio, los controles de selección de entrada y los controles de visualización de vistas adicionales. Éstos últimos se limitan a activa o desactivar la visibilidad de las vistas adicionales. En cambio, los controles de selección de entrada realizan una labor mucho más importante, ya que permiten entre otros usos, escoger si se usarán datos reales leídos de una base de datos, o si por el contrario, se emplearán funciones para simular la obtención de estos datos.

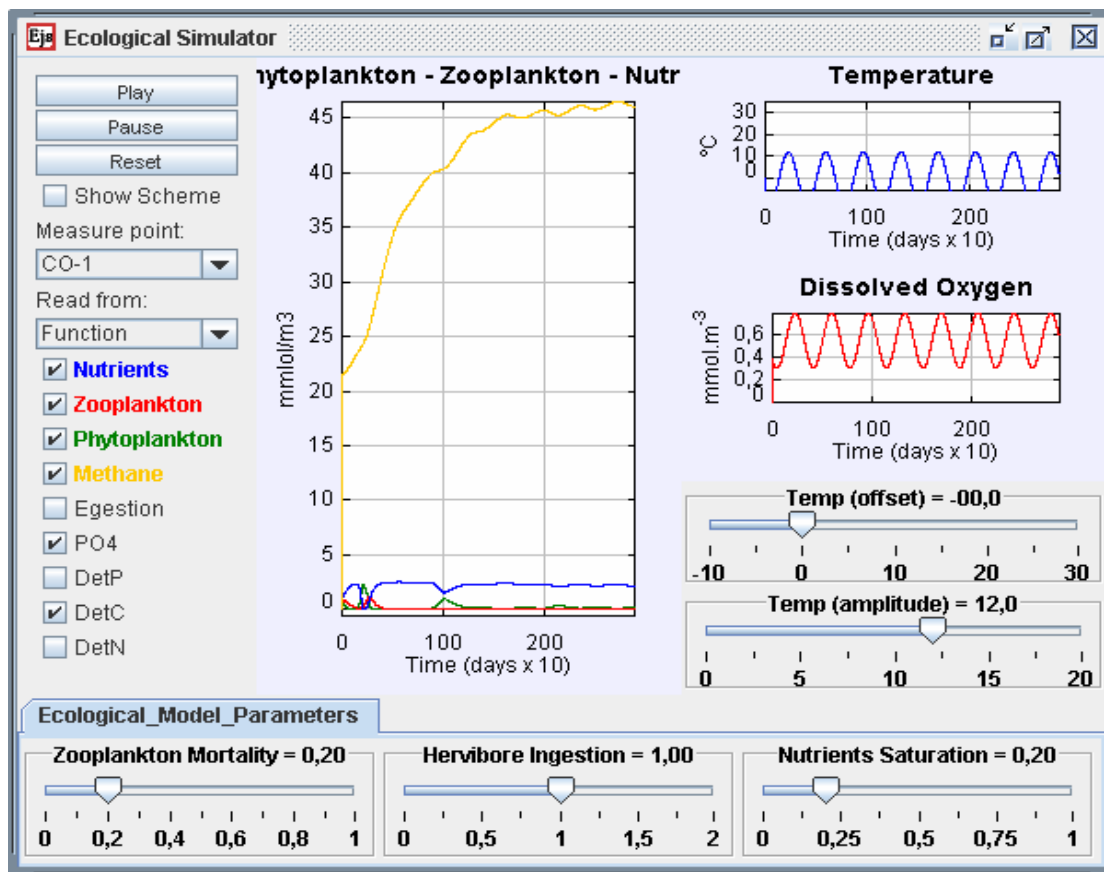


Fig. 17: Vista principal del modelo PZNPO+Carbon

En la zona central de la vista principal se observa una gráfica con la evolución de las principales variables de estado y salida, a saber: nutrientes, zooplancton, fitoplancton y metano emitido; siendo posible activar/desactivar la visualización de las trazas de cada variable. El resto de la vista principal permite controlar y/o visualizar los distintos parámetros y variables de entrada. Así, mediante sendas gráficas, se puede ver la evolución del oxígeno y de la temperatura, y siendo el caso de realizarse la simulación sin acceso a una base de datos, es posible controlar el comportamiento de la temperatura fijando su media y la desviación máxima mediante los deslizadores de control. También mediante deslizadores de control es posible controlar algunos parámetros como la mortalidad del zooplancton entre otros.

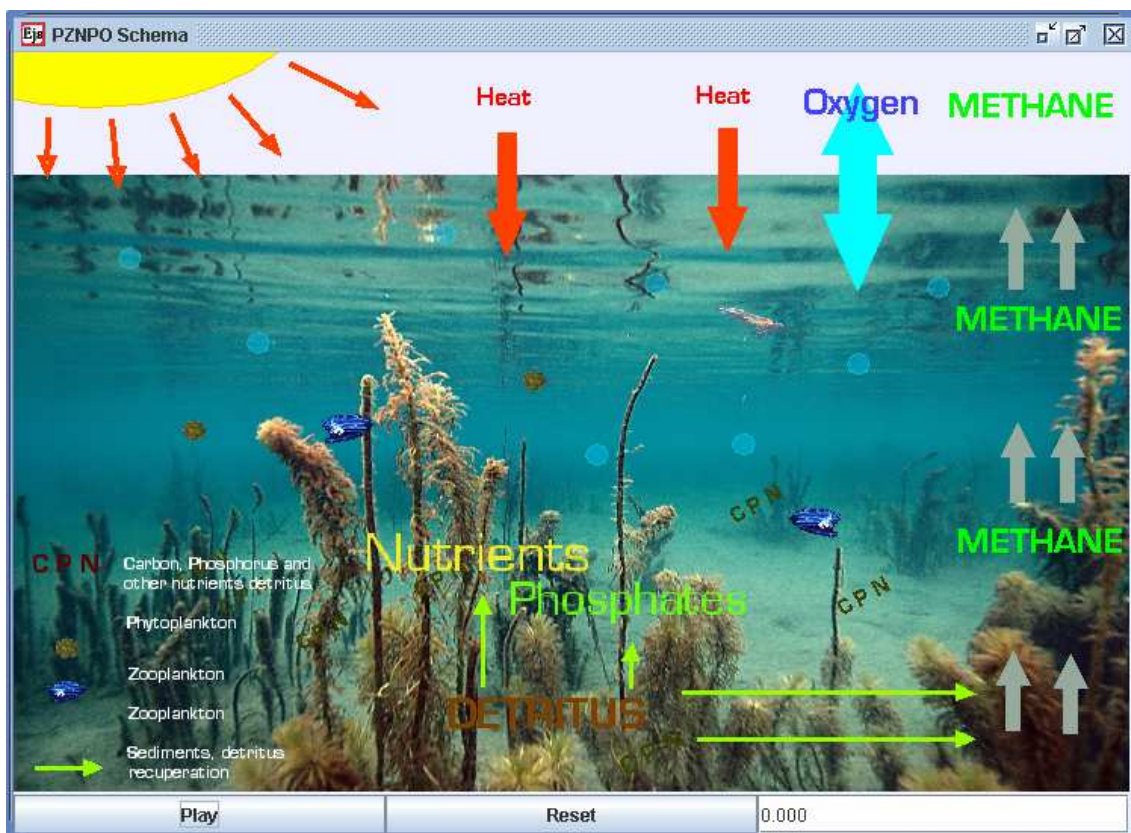


Fig. 18: Esquema animado del modelo PZNPO+Carbon

Del conjunto de vistas adicionales, la más destacable es el esquema animado (figura 18). Uniendo el uso de elementos visuales realizados con EJS, tanto fijos como dinámicos, e imágenes simples y animadas, se ha implementado una representación dinámica del esquema de flujo del modelo. Así, es posible observar como los detritos de *C*, *P* y *N* decaen en cantidades variables, como las poblaciones de fitoplancton y zooplancton evolucionan en número, y como los distintos compartimientos y flujos

crecen o decrecen de acuerdo a la evolución del modelo. Se ha incluido además una pequeña leyenda para aclarar el significado de los dibujos y pictogramas empleados.

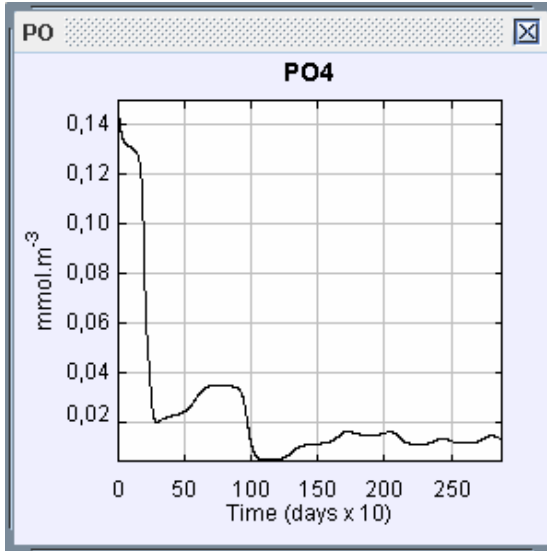


Fig. 19: Vista adicional de ejemplo

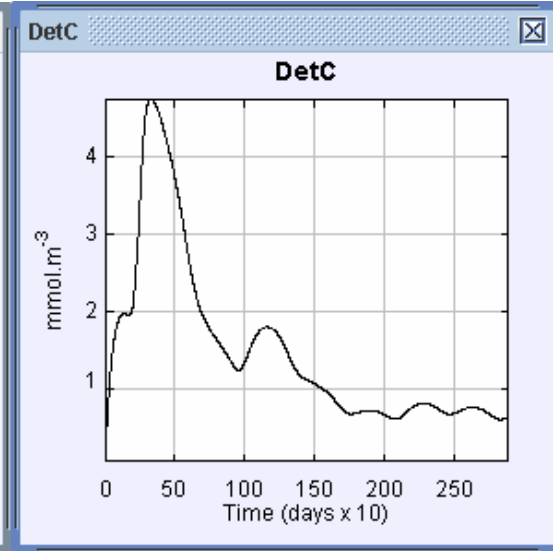


Fig. 20: Vista adicional de ejemplo

Por último, las vistas restantes reflejan la evolución del resto de elementos y compartimentos del modelo de forma individualizada, dada la enorme disparidad de escalas, como se puede ver en los ejemplos de las figuras 19 y 20, dando lugar a una diferencia de hasta dos órdenes de magnitud completos.

8. Modelado de autodepuración de aguas de emisario submarino en campo lejano

8.1. Introducción

Los mares y océanos son indiscutiblemente uno de los mayores recursos ecológicos del planeta Tierra. Producen oxígeno, regulan el clima del planeta, y son usados como fuente de alimento por el hombre. Poseen además una gran capacidad de asimilación, y es por ello que uno de los usos más comunes que hace el hombre de ellos es como receptores de desechos y residuos, especialmente, aguas residuales. Aunque precisamente es este uso como vertedero el que más hace peligrar la biodiversidad y el poder genético de los mares. Siendo así, resulta de vital importancia el estudio del tratamiento y vertido de aguas residuales al mar, con el fin de minimizar su impacto, permitiendo la continuidad de la actividad humana tal y como la conocemos sin hacer peligrar un recurso tan valioso como los mares.

Un hecho destacable, es que independientemente del tratamiento que hayan recibido las aguas antes de ser vertidas al mar, el simple vertido, hecho bajo circunstancias adecuadas, puede contribuir a la depuración de las mismas. Así, tomando medidas adecuadas, se puede minimizar el impacto de los vertidos en el litoral, y hacer que sean de menor impacto para las aguas marinas en general.

De acuerdo con esto, la simulación propuesta se centra en estudiar como distintas variables ambientales afectan a los procesos de depuración de aguas residuales durante el vertido. Simulando así como un emisario submarino actuaría en una zona dada con ciertas variables ambientales conocidas de antemano, evaluando cuales son más favorables para los procesos de depuración.

8.2. Descripción del modelo matemático

El modelo estudia la depuración producida en las aguas vertidas en el mar por un emisario submarino. Un emisario submarino es esencialmente una larga tubería usada para verter aguas residuales, de forma que los contaminantes presentes sean diluidos en la cantidad máxima posible de agua marina, reduciendo el impacto ambiental. La tubería normalmente recorre por el fondo marino una distancia de entre 2000m y 3000m, alejándose de la costa, por donde suele discurrir soterrada. Al final del emisor

suelen hallarse las bocas de vertido o difusores, que conforman el tramo difusor. Este tramo difusor suele tener una longitud del 10% de la longitud total del emisario (figura 21). En la práctica, la mayoría de los emisarios submarinos actuales vierten al mar aguas residuales tratadas al menos parcialmente, ya que las normativas y legislaciones actuales son cada vez más estrictas con la protección de los mares y océanos.

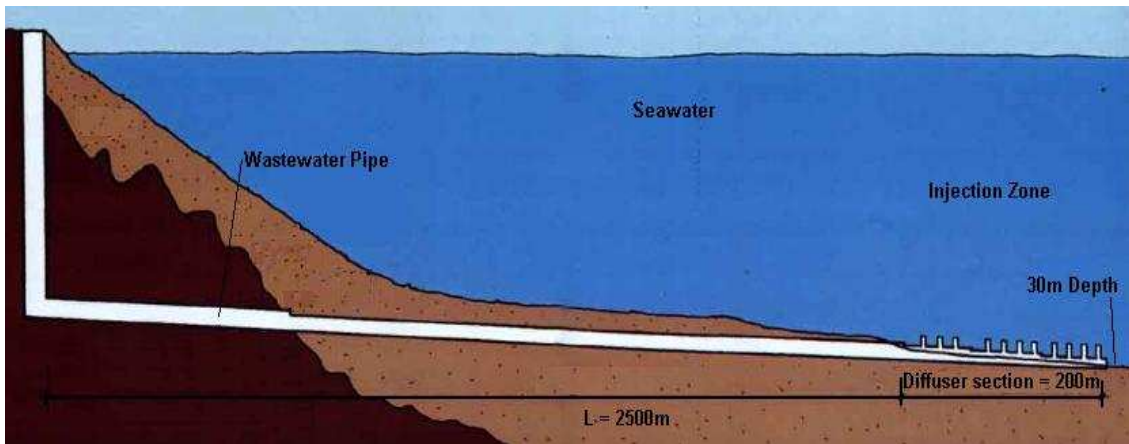


Fig. 21: Recorrido de un emisario submarino

Se pone énfasis específico, dentro de la depuración producida por el vertido a través del emisario, en la fase de difusión turbulenta en el campo lejano. Esto permitirá el estudio de la concentración máxima de contaminante, C_{max} , a una distancia x dada del punto de emisión, así como el ancho del penacho de difusión, b_x . Así se tiene que a una distancia x de la zona de mezcla, se ha formado un penacho de difusión de anchura b_x donde la concentración máxima de contaminante coincide con C_{max} .

Para el modelo de nuestro caso [ESP00], consideramos un emisario submarino de una longitud de 2500m, con una profundidad de vertido de 30m durante el tramo difusor. Este tramo difusor tiene 200m de largo, disponiendo de 20 bocas de vertido o difusores de 0.14m de diámetro, separadas 10m entre ellas. Geográficamente el emisario se sitúa en el Mediterráneo, hecho que definirá variables ambientales como las temperaturas, incidencia solar... aunque sean estos valores modificables por el usuario.

El modelo se centra en el estudio de C_{max} en el campo lejano -dilución secundaria, debida a la difusión en corrientes turbulentas- y la autodepuración bacteriológica -

depuración terciaria, producida por acción bacteriana hostil y la radiación solar, T_{90} en [MES02]-. Con este fin se simula el penacho de dilución de aguas residuales en aguas marinas, considerando múltiples condiciones y variables ambientales modificables.

No obstante, para alcanzar la fase de campo lejano, las aguas residuales han pasado previamente, y por tanto sido diluidas, por la zona de inyección y el campo cercano - dilución primaria, debida a la mezcla turbulenta inicial de aguas residuales y marinas-.

Los distintos factores y variables ambientales relevantes considerados para el modelo son los siguientes:

- Concentración inicial de contaminante, $C_0 = 1$ [ratio]
- Coeficiente de dilución de zona de mezcla, $S = 250$ [sin unidades]
- Velocidad de aguas residuales debida a corrientes y fuerzas de advección ambientales, $U_a = 0$ a 5 [m/s]
- Longitud de la mancha de vertido en superficie marina, $x = 0$ a 4000 [m]
- Amplitud de la mancha de vertido en superficie en zona de mezcla, $B = 0$ a 300 [m]
- Altura de sol, S_a , en grados [°]
- Razón de cielo cubierto por nubes, $C_c = 0$ a 1 [ratio]
- Concentración de sólidos en suspensión, $S_s = 0$ a 800 [mg/l]
- Temperatura del agua, T_w [°C]

Con el fin de simplificar el modelo, los cálculos asociados las diluciones previas en el campo cercano y aquellas debidas a la difusión vertical son ignorados, calculándose únicamente la dilución debida a corrientes turbulentas y la autodepuración bacteriológica. De este modo, conociendo la amplitud de la mancha sobre la zona de mezcla (B), así como otras variables ambientales, se calcula la C_{max} diluida a una distancia x . Para calcular esta C_{max} , se considera un coeficiente medio de dilución de zona de mezcla (S), ya que el penacho surge de esta zona de mezcla, así que sus efectos de dilución se han producido. Posteriormente, contemplando principios de conservación de materia, y considerando advección horizontal a velocidad constante U_a , es posible formar una ecuación, corregida y ajustada gracias a datos estadísticos y experimentales. Básicamente, el crecimiento del penacho hace crecer los coeficientes de difusión turbulenta, ya que la formación de turbulencias de las que dependen aumenta.

U_a también aumenta la turbulencia, intensificando el proceso de dilución turbulenta. Pero al mismo tiempo, reduce el proceso de autodepuración bacteriológica, ya que los elementos contaminantes recorren la distancia x desde la zona de mezcla hasta el campo lejano más rápido, reduciendo el tiempo de exposición a la radiación solar y mitigando el efecto de ésta.

Por tanto, el modelo representante del sistema descrito sería:

$$C_{max}(x) = \frac{C_0}{S} e^{-\frac{\lambda x}{U_a}} \operatorname{erf} \sqrt{\frac{3/2}{\left[1 + \left(8K_y x / U_a B^2\right)\right]^3 - 1}}$$

$$b(x) = B \sqrt{1 + \frac{8K_y x}{U_a B^2}}$$

where

$$K_y = 0.00487 B \frac{4}{3}$$

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

$$\lambda = \frac{\ln 10}{T_{90}}$$

$$T_{90} = \left[\frac{S_a}{60} \left(1 - 0.65 C_c\right) \left(1 - \frac{S_s}{800}\right) + 0.02 \cdot 10^{(T_w - 20)/35} \right]$$

Con el siguiente diagrama de bloques describiendo su comportamiento:

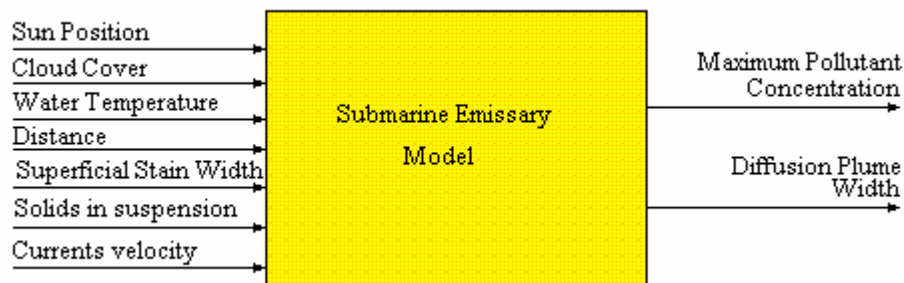


Fig. 22: Diagrama de bloques del modelo

8.3. Control y Visualización

A la hora de implementar las vistas y mecanismos de control para la simulación de este modelo se han mantenido dos máximas como prioritarias: ofrecer una visualización lo más efectiva posible, y mantener la compatibilidad con navegadores en baja resolución. Con este fin se ha dividido la visualización en dos vistas: la vista principal (figura 23), enfocada a el control y una visualización científica, y la vista de apoyo visual (figura 24), con los elementos de control indispensable y una potente simulación tridimensional del penacho de difusión turbulenta formado por el flujo de aguas residuales.

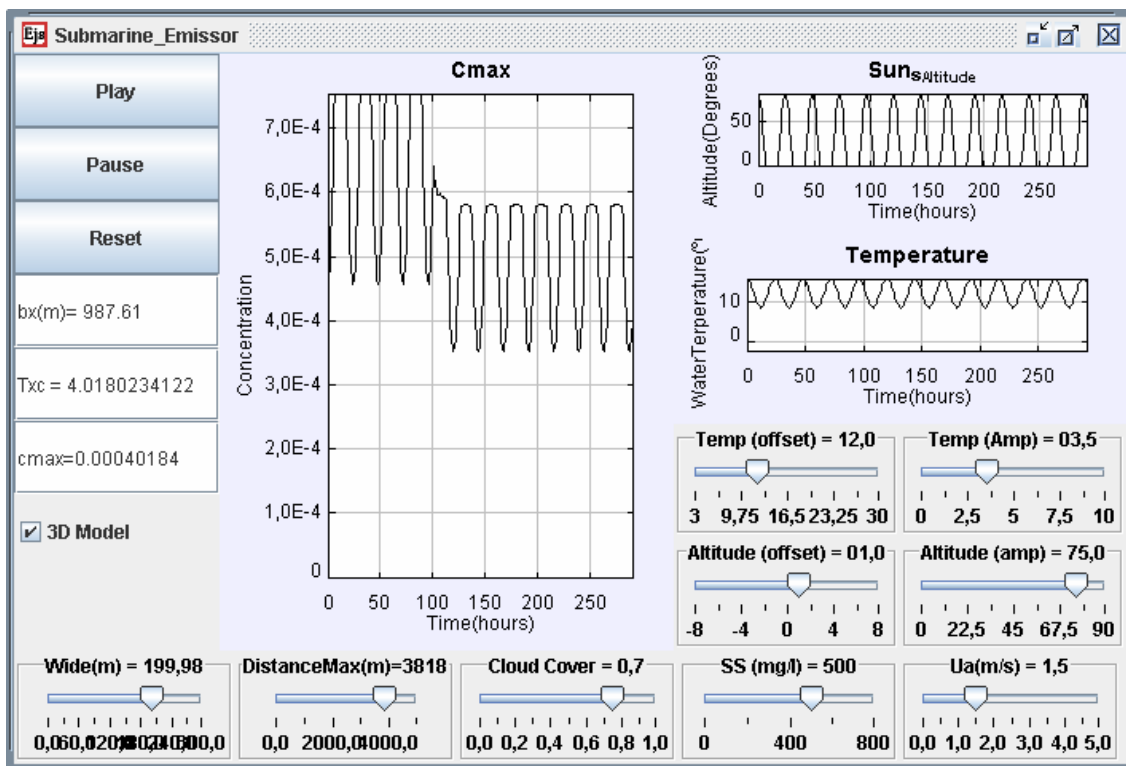


Fig. 23: Vista principal del modelo de emisario submarino

La vista principal dispone de todos los elementos de control de la simulación, aunque algunos estén presentes también en la vista del modelo tridimensional. En primer lugar, se dispone de los controles de ejecución habituales, en la columna izquierda, así como de los campos numéricos para mostrar los valores de b_x , T_{90} y C_{max} en la distancia x fijada, y el casillero para activar la visualización del modelo tridimensional. La gráfica central muestra la evolución de C_{max} en función del tiempo en horas, mostrando un comportamiento cíclico característico, mientras que las otras dos

gráficas muestran respectivamente la altitud del sol en grados y la temperatura del agua.

En cuanto a elementos de control propios del modelo, es posible controlar la mayoría de parámetros y variables mediante deslizadores. Así, inicialmente podemos definir un ancho de mancha visible en la superficie de la zona de mezcla –*Wide*–, y la distancia del punto de emisión que nos interesa evaluar. Tanto la densidad de sólidos en suspensión (S_s), como los valores ambientales, cobertura de nubes (C_c) y velocidad de advección y corrientes (U_a) también son ajustables, para poder simular cambios meteorológicos.

Como se comentó anteriormente, para la realización del modelo se consideró un emisario submarino típico situado en el Mar Mediterráneo. No obstante, para poder simular localizaciones con medias de temperatura y horas de luz diarias diferentes, se pueden controlar la evolución de estas variables mediante deslizadores. De esta forma, se puede variar la temperatura media y la varianza. Para controlar la incidencia de luz solar, se puede definir tanto el ángulo máximo de altura del sol, como ajustar la desviación en la cantidad de horas de día respecto a una media de doce horas -el deslizador de *Altitude(offset)* nos permite sumar o restar hasta 8 horas de luz diarias medias-.

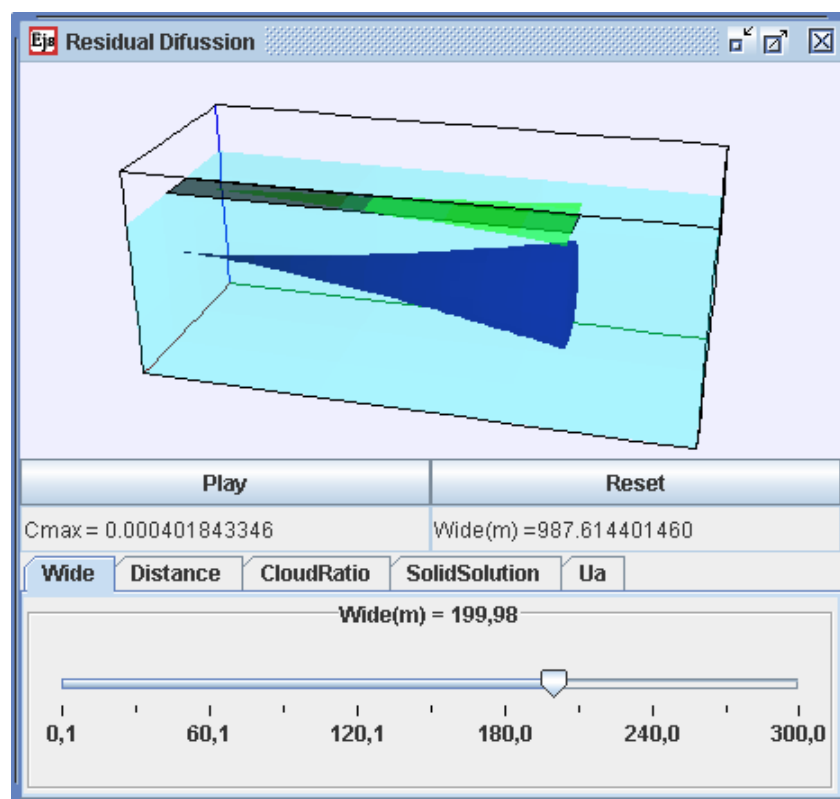


Fig. 24: Vista de apoyo con simulación 3D del penacho

La vista adicional, activada mediante el casillero *3DModel*, muestra una simulación tridimensional del penacho de difusión turbulenta que se formaría a partir de la zona de mezcla en el punto de emisión. El penacho, dividido en 10 zonas, varía la luminosidad del tono de azul en función del nivel de contaminación presente. Es decir, a mayor C_{max} , esa zona del penacho se verá de un tono más oscuro y contaminado. El ancho del penacho también es proyectado verticalmente hacia la superficie, mientras que la amplitud de la mancha visible sobre la zona de mezcla es arrastrada por la superficie. Esta combinación permite visualizar y ofrecer una idea de la amplitud del área submarina que puede haber contaminada viéndose una zona superficial contaminada relativamente pequeña.

En cuanto a elementos de control, esta última vista dispone de los controles de ejecución y deslizadores de los principales parámetros y variables del modelo, con la excepción del control de temperatura y altitud solar. Estos controles se han omitido deliberadamente, dado que generalmente despertaran interés en usuarios más interesados en los datos científicos de la vista principal que no en la simulación del penacho.

9. Del Caos al Orden y viceversa en ecosistemas acuáticos

9.1. Introducción

La presencia del caos en la naturaleza y sus ecosistemas ha sido un punto de polémica y debate para la comunidad científica durante años. Dado que la principal característica del caos es la ausencia de un punto o ciclo estable de equilibrio, el caos resulta difícil de estudiar en la naturaleza, siendo un punto frecuente de discusión como caracterizar su presencia. Aún así, estudios relativamente recientes sobre dinámicas de sistemas han considerado las fluctuaciones caóticas como altamente deseables, dado que el sistema se vuelve más fácil de controlar y regular.

Este conocimiento sobre el caos en la naturaleza se ha aplicado en diferentes ámbitos, siendo uno de ellos el estudio de redes tróficas. A un nivel teórico resulta muy interesante estudiar como puede un ecosistema autorregularse y buscar el equilibrio. Sin embargo, desde un punto de vista más pragmático, llegar a comprender los mecanismos de autorregulación de ciertos ecosistemas, como funcionan y que les hace dispararse es de una importancia crítica. Como veremos en el modelo simulado, uno de los mecanismos más usuales de autorregulación de las redes tróficas viene siendo la autodefensa de una especie contra otra. Y cuando para su defensa contra la depredación del zooplancton, el fitoplancton secreta toxinas altamente perniciosas, hay un problema ecológico, de salud humana, y producción alimenticia que necesita ser estudiado.

9.2. Descripción del modelo matemático

Este sistema modela el comportamiento de una cadena trófica de tres especies en un ecosistema acuático. En este ecosistema tienen lugar dinámicas caóticas y se pueden estudiar los mecanismos de regulación disponibles. Las principales variables estudiadas serán las poblaciones de las tres especies:

1. **Fitoplancton (P)**
2. **Zooplancton (Z)**
3. **Peces (F)**

A través del modelo estudiado [MAN06] es posible ver como pueden surgir situaciones caóticas de un estado de equilibrio por distintas razones, como estas situaciones afectan a la evolución de las 3 especies presentes en la cadena trófica (P , Z , F), y como el ecosistema puede superar esta situación mediante los mecanismos disponibles y su capacidad de adaptación.

La cadena trófica representada refleja las interacciones usuales entre especies acuáticas: el zooplancton (Z) consume fitoplancton (P), y los peces (F) a su vez consumen zooplancton (Z). Estas interacciones están modeladas de acuerdo con el modelo de respuesta funcional “tipo II” de Holling. La cadena trófica de tres niveles es de especial interés dado que en ella se observan dinámicas caóticas que no se darían en modelos de dos especies bajo parámetros y circunstancias biológicamente razonables.

El modelo introduce un proceso de liberación de toxinas por parte de la población de fitoplancton como sistema de autorregulación del ecosistema, el cual reduce la presión de consumo del zooplancton sobre el fitoplancton. Cuando se introduce la interferencia debida a la liberación de toxinas en la relación entre zooplancton y fitoplancton, se observa como respuestas funcionales de tipo I y tipo III pasan a ser necesarias para describir el comportamiento del sistema adecuadamente.

Todas estas consideraciones permiten desarrollar un modelo de ecuaciones diferenciales ordinarias:

$$\frac{dP}{dT} = R_0 P \left(1 - \frac{P}{K_0} \right) - \frac{C_1 A_1 P Z}{B_1 + P}$$

$$\frac{dZ}{dT} = \frac{A_1 P Z}{B_1 + P} - \frac{A_2 Z F}{B_2 + Z} - D_1 Z - f(P) Z$$

$$\frac{dF}{dT} = \frac{C_2 A_2 Z F}{B_2 + Z} - D_2 F$$

donde :

$$f(P) = \frac{\beta P}{B_1 + P}$$

Donde T es el tiempo. La constante R_0 es la tasa de crecimiento intrínseco del fitoplancton, y la constante K_0 es la capacidad de transporte de fitoplancton. C_1 y C_2 son las tasas de conversión de presas a depredadores para el zooplancton y los peces, respectivamente; D_1 y D_2 son las constantes que describen las tasas de mortalidad del zooplancton y los peces. A_n y B_n se $-n = (1, 2)$ - emplean para

parametrizar las respuestas funcionales de tipo II, siendo B_n el nivel de población de presas donde la tasa de depredación por unidad de presa se halla a la mitad de su máximo. $f(P)$ representa el proceso de liberación de toxinas del fitoplancton, en el que β es la tasa de liberación de toxinas.

Ya que en la especificación dada hasta ahora, el modelo consta de once parámetros y variables, para simplificar su análisis y estudio este número es reducido. Para poder reducir el número de variables y determinar cuáles controlan en mayor medida el comportamiento del modelo, se han eliminado las unidades convirtiéndose en un modelo adimensional, realizando las siguientes sustituciones:

$$\begin{aligned} p &= P / K_0 \\ z &= C_1 Z / K_0 \\ f &= C_1 F / C_2 K_0 \\ t &= R_0 T \end{aligned}$$

Estas sustituciones simplifican las ecuaciones, dando lugar al sistema siguiente, donde p , z y f son medidas a escala de tamaño de poblaciones, las cuales son fácilmente implementables como variables reales, y t es una nueva variable de tiempo. Así, p , z , f y t se convierten en variables adimensionales sin unidades:

$$\begin{aligned} \frac{dp}{dt} &= p(1-p) - \frac{a_1 pz}{1+b_1 p} \\ \frac{dz}{dt} &= \frac{(a_1 - \theta) pz}{1+b_1 p} - \frac{a_2 zf}{1+b_2 z} - d_1 z \\ \frac{df}{dt} &= \frac{a_2 zf}{1+b_2 z} - d_2 z \end{aligned}$$

Correspondiéndose las nuevas variables adimensionales con las variables iniciales:

$$\begin{aligned} a_1 &= K_0 / R_0 B_1 \\ a_2 &= A_2 C_2 K_0 / C_1 R_0 B_1 \\ b_1 &= K_0 / B_1 \\ b_2 &= K_0 / B_2 C_1 \\ d_1 &= D_1 / R_0 \\ d_2 &= D_2 / R_0 \\ \theta &= \beta K_0 / R_0 B_1 \end{aligned}$$

Ahora, en el sistema de ecuaciones diferenciales resultante de los cambios, tenemos a_n como la tasa de crecimiento específico, b_n como constante de media saturación y d_n como tasa de mortalidad específica; donde $n=1$ son valores para el zooplancton y $n=2$ indica valores para los peces. θ mide el proceso de liberación de toxinas del fitoplancton, el cual actúa como uno de los mecanismos principales de adaptación del sistema. Así, la simulación permite observar como un incremento progresivo en la tasa de liberación de toxinas desplazará el sistema de un estado caótico a una fase de *periodo doble*, pasando posteriormente por un *periodo de ciclo límite*, antes de regresar a la fase de *punto de equilibrio*.

Con los parámetros y variables ya descritos, el diagrama de bloques que describe el comportamiento del modelo sería:

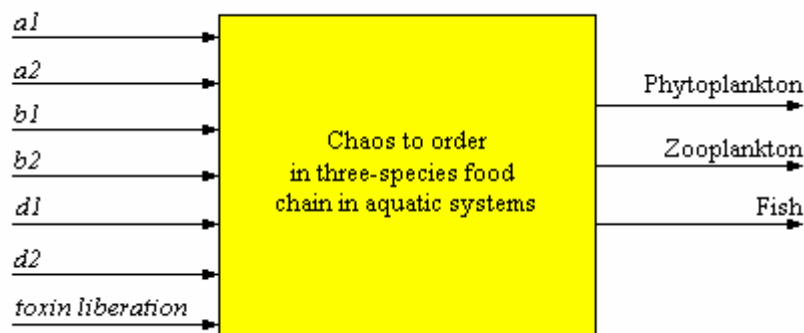


Fig. 25: Diagrama de bloques del modelo

9.3. Control y visualización

La simulación del modelo cuenta con todos los elementos necesarios para controlar todos los parámetros, y observar todas las variables en ejecución. Sin embargo, una serie de limitaciones técnicas del programario EJS ha forzado la inclusión de una vista adicional para controlar adecuadamente ciertos elementos de visualización.

De este modo tenemos una vista principal (figura 26), dividida en: zona de paneles de visualización -*PanelConSeparadores* de EJS- ocupando la parte izquierda y central; evaluación y control de variables y parámetros, en la columna derecha; y la zona de control de ejecución y datos, a lo largo del borde inferior.

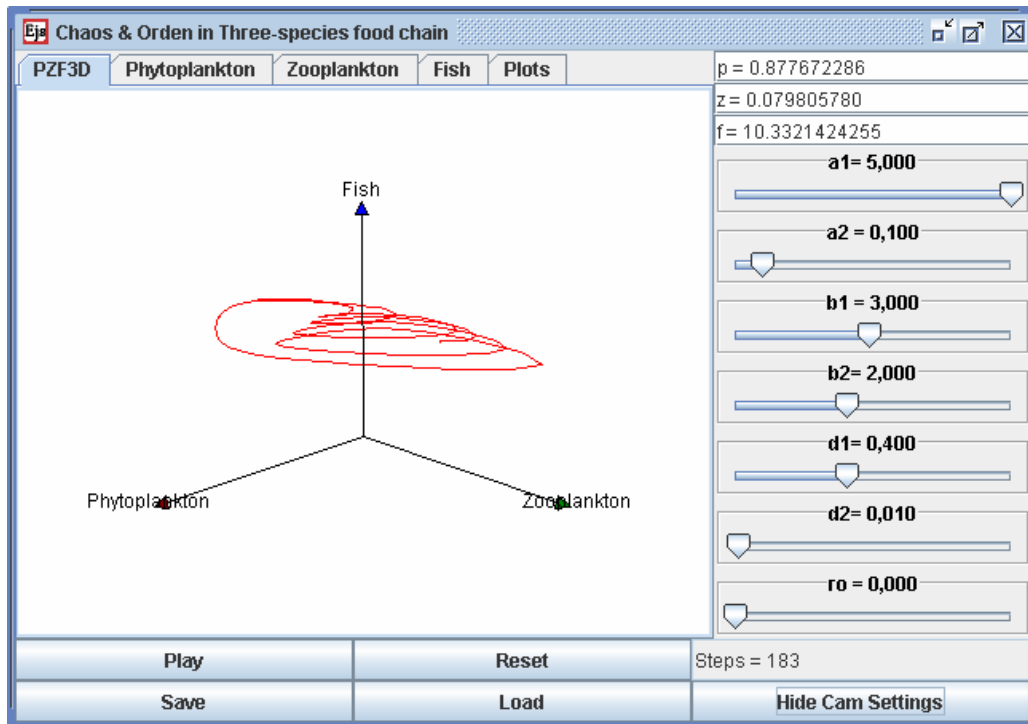


Fig. 26: Vista principal de la simulación, viendo panel con gráfica 3D.

Los controles de ejecución y datos son los habituales botones de iniciar y reiniciar la simulación, con algunos añadidos. El control de salvado de estado -botón *Save*- permite guardar los valores de ejecución de todas las variables excepto el número de pasos de ejecución -recordemos que en el modelo el tiempo es convertido a una magnitud adimensional sin unidades, por tanto de relativo poco interés-. Esta funcionalidad, junto con la opción de cargado de este estado -botón *Load*- permite reiniciar la simulación a partir de ese punto guardado, obteniendo un estado “limpio”, sin datos previos ni trazas en la gráfica dificultando la visualización. Adicionalmente se ofrece el botón para visualizar y controlar los parámetros de cámara de la gráfica 3D ofrecida en los paneles de visualización.

Dentro de los paneles de visualización, el primero de ellos ofrece una gráfica 3D que muestra la evolución de sistema dentro del espacio de estados posibles (figura 26). Una de las limitaciones de EJS ha sido precisamente que a diferencia de las gráficas en 2D habituales, las gráficas en 3D no disponen de un elemento específico para su visualización. Así, mientras una gráfica que relacione 2 variables dispone de un algoritmo de autoescalado que ofrece una vista ajustada óptima, una gráfica se dibuja en un espacio 3D de dibujo de tamaño fijo. Implicando así que no se pueda escalar

automáticamente la gráfica, y dada la enorme disparidad posible en la magnitud de las variables, la traza puede llegar a verse desproporcionada en relación al espacio visual disponible –o no verse en absoluto-. Como medida para paliar este problema, se ha implementado la mencionada vista adicional, que permite controlar los parámetros de la cámara, tal y como se detalla a continuación.

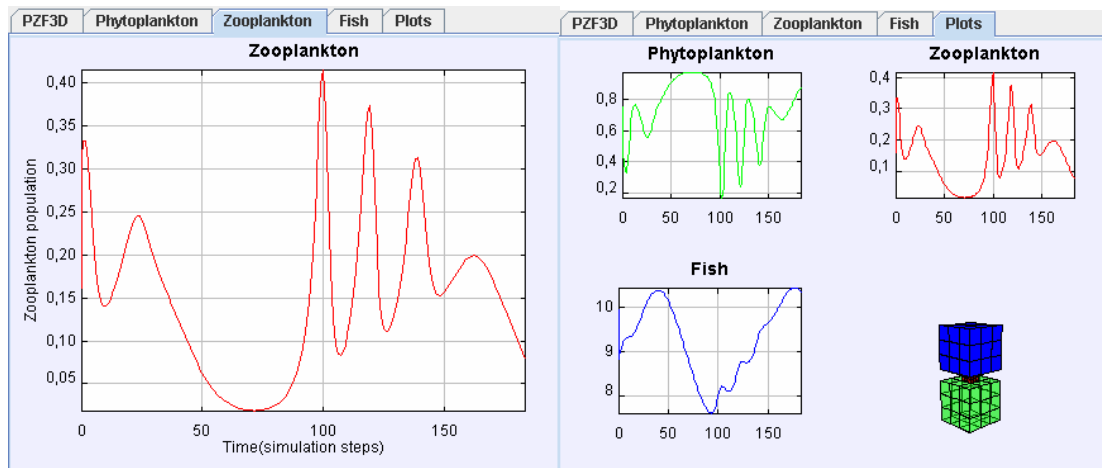


Fig. 27: Vista de un panel de evolución

Fig. 28: Vista de gráficas de evolución juntas

El resto de los paneles de visualización ofrecen gráficas 2D independientes para cada una de las variables (p , z , f) en función del tiempo (figura 27), tanto individualmente, para apreciar la evolución de cada una en detalle, como un panel con las gráficas de las tres variables para estudiar su comportamiento conjunto (figura 28). Este panel además contiene un pequeño “diagrama de cubos” que muestra el crecimiento y decrecimiento de cada variable, tomando la forma de cubos “normalizados”.

La vista adicional, *Cam Settings* (figura 28), permite ajustar la cámara que visualiza la gráfica tridimensional para obtener una mejor visualización y detalle. Controla esencialmente la posición de la cámara y su punto de enfoque en relación a los ejes. También permite cierto control sobre el *zoom* y la rotación horaria de la cámara - observándose la rotación antihoraria de la gráfica y sus ejes-.

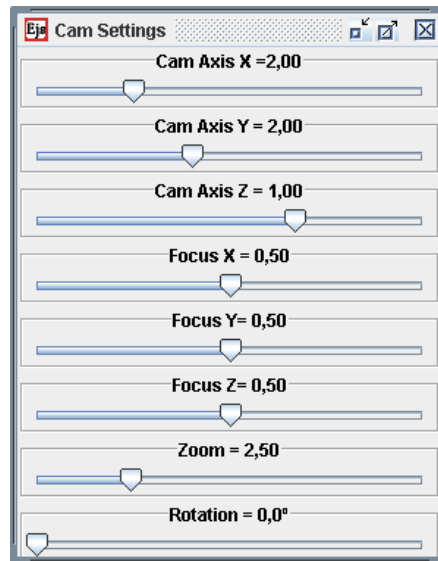


Fig. 29: Vista de los controles de la cámara 3D

9.4. Respuestas funcionales de Holling

El concepto de respuestas funcionales fue introducido por C.S. *Holling*, un ecologista canadiense, en 1959. Se emplea para describir la relación entre la densidad de presas en un área determinada, y el número de presas promedio consumido por cada depredador en esa área. También a la vez introdujo el concepto de respuesta numérica, el cual describe la relación entre la densidad de presas y la densidad de depredadores. Estos conceptos aún son considerados clave en el modelado de sistemas ecológicos.

Con el fin de describir adecuadamente las distintas relaciones entre la densidad de presa y la presión de depredación por parte de los depredadores, Holling detalló tres tipos de respuestas funcionales [URL10]:

Tipo I (*lineal*): es la respuesta funcional de los depredadores pasivos, quienes consumen presas tal y como las van encontrando, sin buscarlas activamente. El número de presas consumidas por un depredador aumenta proporcionalmente a la densidad de presas hasta que está saciado. A partir de ese punto se vuelve constante (figura 30).

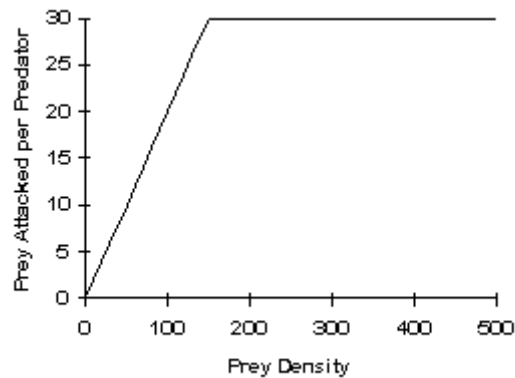


Fig. 30: Respuesta de Holling tipo I

Tipo II (*cirtoide*): es la respuesta funcional más típica y corresponde a una especie depredadora especializada en una o varias presas. El esfuerzo de búsqueda de presas es constante, así que la tasa de mortalidad es máxima con densidad de población de presas baja. La tasa de ataque a presas disminuirá conforme aumente la densidad de población de presas, hasta que el depredador esté saciado, cuando se volverá constante (figura 31).

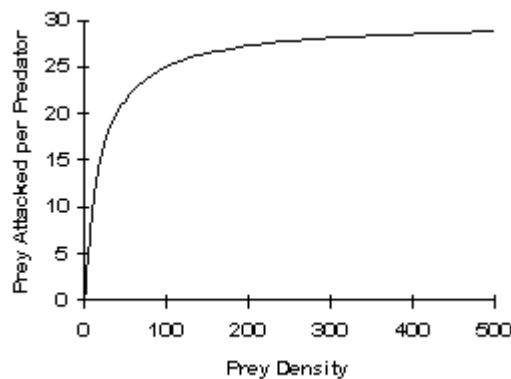


Fig. 31: Respuesta de Holling de tipo II

Tipo III (*sigmoide*): las respuestas funcionales de este tipo se dan en depredadores que incrementan su actividad de búsqueda cuando incrementa la densidad de presas. Eventualmente el tiempo de actividad de búsqueda decrecerá, al aproximarse al saciado. Esta respuesta es típica en depredadores generalistas que rápidamente

cambian de una especie de comida a otra y concentran su alimentación en áreas donde ciertos recursos sean abundantes (figura 32).

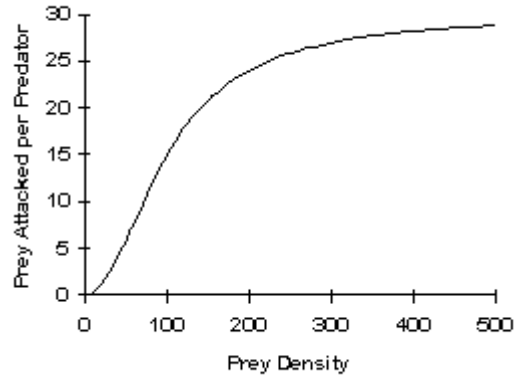


Fig. 32: Respuesta de Holling de tipo III

9.5. Fases de estado del sistema

La simulación presenta inicialmente el sistema modelado en un *estado caótico* (figura 33), permitiendo la variación de distintos parámetros para ver los efectos en el sistema. Los parámetros con mayor impacto en el sistema son los relacionados con el zooplancton, especialmente la constante de media saturación del zooplancton (b_1) y la tasa de liberación de toxinas (θ), ya que ambos controlan y/o limitan el crecimiento de éste.

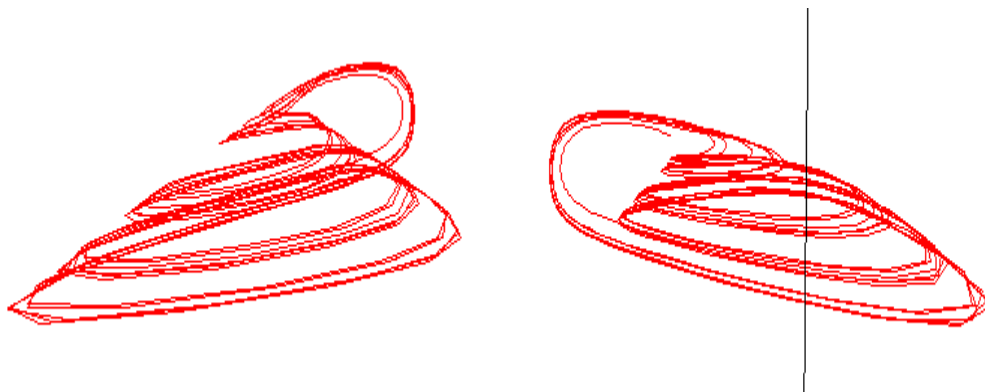


Fig. 33: Ejemplos de estado caótico

Empezando desde un *estado caótico*, para alcanzar el *punto de equilibrio*, el sistema normalmente cambiará a una breve *fase de doble periodo* (figura 34):



Fig. 34: Ejemplo de fase de doble periodo

En esta *fase de doble periodo*, el sistema muestra un patrón vagamente similar al del *estado caótico*. Pero mientras en el estado caótico cada pasada de las curvas creaba una traza nueva, en la fase de *doble periodo* el sistema recorre alternativamente una de dos trazas posibles. Cualquier variación de los parámetros que acerque el sistema al punto de equilibrio pasará inmediatamente antes a *una fase de ciclo límite* (figura 35), donde se observa un comportamiento cíclico claro:

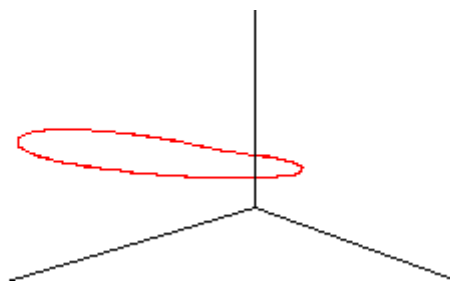


Fig. 35: Ejemplo de ciclo límite

El sistema podría permanecer en esta fase indefinidamente, pero si continúa la variación, el sistema entrará en fase de "*retorno al punto de equilibrio*" (figura 36). Esta fase es fácilmente reconocible por su característica aproximación espiral al posible *punto de equilibrio*.

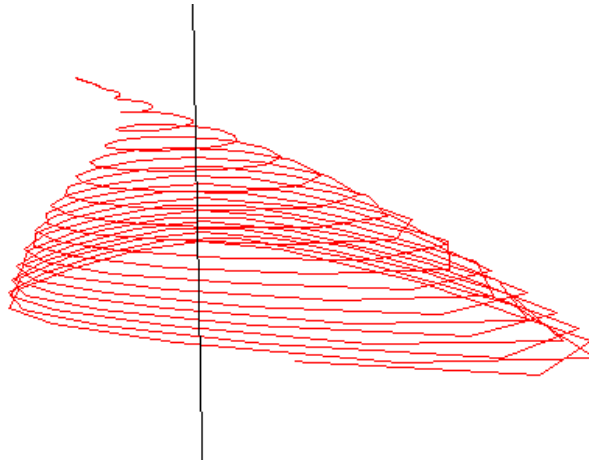


Fig. 36: Ejemplo de retorno al punto de equilibrio

En algunos casos, el sistema no consigue hallar el *punto de equilibrio* esperado, dando lugar a una nueva fase (figura 37) con el característico movimiento en espiral:

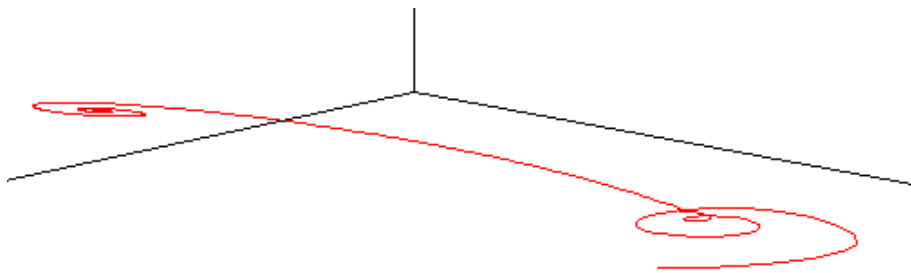


Fig. 37: Intento fallido de retorno al punto de equilibrio

Al estudiar el proceso opuesto, emergencia de *estados caóticos* desde un *punto de equilibrio*, las fases tienen orden lugar en un orden inverso. Así, normalmente, del *punto de equilibrio* se llega a una fase de *ciclo límite* tras recorrer una espiral. Después se pasaría por la fase de *periodo doble*, hasta llegar al *estado caótico*.

10. Modelado de flujo de dos fases del drenaje de una veta de carbón

10.1. Introducción

Un ámbito donde es crítico el estudio de la sostenibilidad es en las actividades industriales de cualquier tipo. Indispensables en la concepción del mundo actual, estas actividades han sido tradicionalmente muy perniciosas para el medio ambiente. Dada la enorme dependencia de la actividad humana de los procesos industriales, resulta de especial interés estudiarlos para evaluar y mitigar su impacto.

De entre las actividades industriales, la minera ha sido siempre de las más vinculadas al auge y caída de imperios. Ya fuesen los de la antigüedad, forjados con las distintas armas de bronce, hierro y acero o comprados con oro y plata, u otros más recientes, como los económicos nacidos gracias al carbón que impulsó la revolución industrial, la mayoría de imperios deben mucho a la minería. No obstante esta actividad aún hoy sigue siendo relativamente peligrosa en comparación con otras industrias primarias, además de muy contaminante.

Por ello resulta interesante estudiar como las actividades mineras pueden afectar al medioambiente de formas diversas. Desde la emisión de gases y partículas contaminantes hasta los efectos del drenaje de aguas en las aguas subterráneas colindantes, hay muchos aspectos a estudiar en este tipo de actividades.

El modelo presentado se centra en como evoluciona la presión de agua y metano en el subsuelo, a raíz de la perforación de un agujero de drenaje para permitir posteriormente la actividad minera. Dado que es un proceso lento, interesa disponer de mecanismos que permitan “acelerar” la simulación, o permitir obtener los datos de interés mientras se ejecuta en *background* por ejemplo. Como en otros modelos, también será interesante mostrarla de alguna forma más visual y atractiva.

10.2. Descripción del modelo matemático

El modelo está diseñado para prever y simular el comportamiento de un flujo de dos fases, de gas metano y de agua, durante las operaciones de drenaje de para una veta

de carbón en Bulli, Australia. Con este fin se estudia el comportamiento de una serie variables, a saber:

1. **Presión de agua (P_w)**
2. **Presión de gas (P_g)**
3. **Presión capilar (P_c)**

Así, a través del modelo [BAS88], es posible estudiar la evolución de tanto la presión de agua como la de gas, en cuatro profundidades diferentes, a distintas distancias de un punto central de drenaje. Para simplificar, se ha asumido un flujo unidireccional en la veta de carbón. Esta veta ha sido modelada considerándola con propiedades isotrópicas del carbón, como la porosidad y la permeabilidad. Asimismo se ha considerado incompresible, isotérmica y totalmente horizontal. También se ha considerado incompresible el flujo del agua, y el metano con comportamiento de gas ideal. Los flujos a través de la veta se asume que son de tipo Darcy laminar con capilaridad de dos fases completa y efectos de permeabilidad.

Teniendo en cuenta todas estas premisas y el entorno de la mina de carbón en Bulli, se han definido una serie de parámetros y condiciones previas:

- Temperatura media en la entrada de la mina = 22 [°C]
- Viscosidad del agua, $\mu_w = 1.0 \cdot 10^{-3}$ [Pa·s]
- Porosidad, $\phi = 0.08$ [constante]
- Permeabilidad absoluta, $k = 80 \cdot 10^{-3}$ [Darcy]
- Densidad del agua = 1000 [kg/m³]
- Densidad del gas = 0.52 [kg/m³] a 80 [kPa]
- Constante del gas metano, $R = 518$ [J/kg·K]

El modelo resultante es un conjunto de complejas ecuaciones parciales, de considerable dificultad:

$$\frac{\partial p_g}{\partial t} = \frac{k \cdot k_{rg}}{2\phi\mu_g(1-s_w)} \left[\frac{\partial^2 p_g^2}{\partial x^2} \right] + \frac{k \cdot k_{rg} p_g}{2\phi\mu_g(1-s_w)} \left[\frac{\partial^2 p_w}{\partial x^2} \right]$$

$$\frac{\partial p_w}{\partial t} = \frac{k \cdot k_{rg}}{2\phi\mu_g(1-s_w)} \left[\frac{\partial^2 p_g^2}{\partial x^2} \right] - m \frac{k \cdot k_{rw} p_g}{2\phi\mu_g(1-s_w)} \left[\frac{\partial^2 p_w}{\partial x^2} \right]$$

donde

$$m = \frac{1-s_w}{p_g} \frac{dp_c}{ds_w} - 1$$

Por ello, con el fin de evitar la complejidad subyacente a las técnicas de cálculo analítico, estas ecuaciones de flujo son resueltas numéricamente empleando formulación de diferencias explícita. Este procedimiento se puede realizar considerando los datos y condiciones iniciales obtenidos experimentalmente, y a condición de no haber flujo en el límite axial. De esta manera es posible reemplazar las complejas ecuaciones con diferenciales parciales por aproximaciones de diferencias explícitas:

$$p_g(x, t + \Delta t) = \frac{k \cdot k_{rg}}{2\phi\mu_g(1-s_w)} \frac{\Delta t}{\Delta x^2} [p_g^2(x + \Delta x, t) - 2p_g^2(x, t) + p_g^2(x - \Delta x, t)]$$

$$+ \frac{k \cdot k_{rw} p_g(x, t)}{\mu_w(1-s_w)} \frac{\Delta t}{\Delta x^2} [p_w(x + \Delta x, t) - 2p_w(x, t) + p_w(x - \Delta x, t)] + p_g(x, t)$$

$$p_w(x, t + \Delta t) = \frac{k \cdot k_{rg}}{2\phi\mu_g(1-s_w)} \frac{\Delta t}{\Delta x^2} [p_g^2(x + \Delta x, t) - 2p_g^2(x, t) + p_g^2(x - \Delta x, t)]$$

$$- \frac{k \cdot k_{rw} p_g(x, t)}{\mu_w(1-s_w)} \frac{\Delta t}{\Delta x^2} m [p_w(x + \Delta x, t) - 2p_w(x, t) + p_w(x - \Delta x, t)] + p_w(x, t)$$

donde

$$m = \frac{1-s_w}{p_g} \frac{dp_c}{ds_w} - 1$$

Así, el modelo dispone de múltiples variables de estado para representar P_g y P_w en una cantidad discreta de puntos. Estos puntos están repartidos a cuatro niveles de profundidad, 0m, 10m, 20m y 30m, y se separan longitudinalmente de agujero de drenaje de metro en metro. Es imprescindible tener un conjunto de valores iniciales para P_g y P_w en todos los puntos de cada capa - $p_g(x, 0)$ y $p_w(x, 0)$ -, pues, partiendo de estos valores, el modelo estima los siguientes valores de P_g y P_w como las funciones

$\rho_g(x,t+\Delta t)$ y $\rho_w(x,t+\Delta t)$, avanzando la simulación un paso de tiempo. Pese a que la formulación del modelo emplea como unidad básica de tiempo el segundo, siendo ésta la resolución óptima, es enteramente posible definir un periodo de tiempo mayor como el transcurrido en cada paso de simulación. Esto permite estudiar periodos más largos de tiempo -como meses-, aunque tampoco es una práctica aconsejable, ya que el comportamiento del modelo se vuelve progresivamente impreciso, e incluso puede llegar a ser inestable e inservible -a partir de pasos de ejecución del orden de 1000 segundo-.

$$P_c = P_g - P_w$$

El término presión capilar (P_c) describe la diferencia de presiones entre el gas y el agua en la interfaz de su unión, curvada debido a los efectos de la tensión superficial. El valor de la presión capilar se emplea para calcular el nivel de saturación de agua (W_s), gracias a una función definida en trabajos previos basados en datos experimentales. Una vez estimada la saturación del agua, ésta nos permite calcular las permeabilidades relativas del agua y el gas, k_{rw} y k_{rg} respectivamente, siendo sus valores función de W_s de acuerdo a trabajos previos, como muestran las gráficas (figura 38).

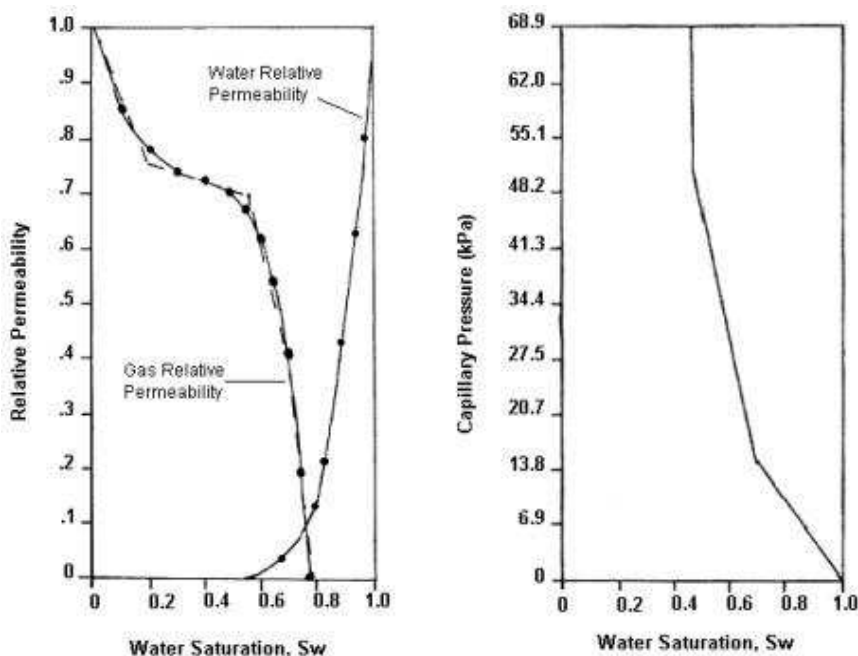


Fig. 38: Relaciones de P_c con W_s , y de W_s con k_{rw} y k_{rg}

Los datos experimentales empleados se basan en otros trabajos previos y en información obtenida durante los experimentos llevados a cabo en West Cliff Colliery, Australia (tabla 3). Durante la realización de estos, se perforó inicialmente un agujero de drenaje de 24m de profundidad, seguido de seis perforaciones de 30m de profundidad dispuestas en línea a 4, 6, 9, 16, 25 y 35m del agujero de drenaje. En cada una de las 6 perforaciones adicionales se dispusieron los dispositivos necesarios para tomar mediciones de las distintas presiones a profundidades de 0, 10, 20 y 30 metros, arrojando los siguientes resultados:

Depth into Borehole	Distance of borehole from draining hole						
	0 m	4 m	6 m	9 m	16 m	25 m	35 m
0 m	80	80	80	80	80	80	80
10 m	80	-	-	93	96	99	108
20 m	80	140	-	290	348	-	359
30 m	80	854	854	860	878	892	887

Tabla 3: Presión iniciales en la veta, en kPa

Más allá de los 35m la presión se considera constante, dado que experimentos previos mostraron que el efecto del flujo de agua y gas era negligible. Se estimó que un valor de incremento de distancia (Δx) de 1m arrojaría resultados adecuados. Dado que para la inicialización de los cálculos es necesario tener los valores de $p_g(x,0)$ y $p_w(x,0)$ para todos los puntos, y que Δx se fijó en 1m, fue necesario estimar parte de los valores. Para realizar esta estimación de los valores necesarios se utilizó interpolación por “spline” cúbico [URL11]. Este tipo de interpolación tiene la ventaja de no verse afectado por el fenómeno de Runge pese a usar puntos equidistantes.

El diagrama de bloques que describe el comportamiento del modelo sería el siguiente:

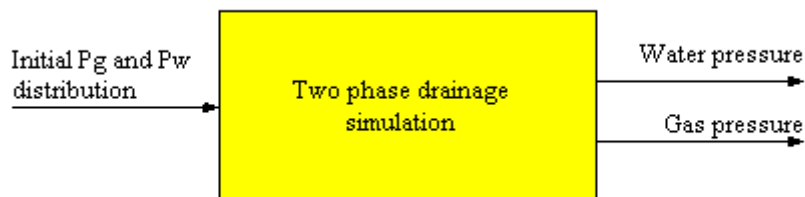


Fig. 39: Diagrama de bloques del modelo

Dada esta caracterización del modelo, las variables de entrada serían el conjunto de valores iniciales de $P_g(x,0)$ y $P_w(x,0)$ para todo x entero de 0 a 36 para las cuatro capas de profundidad consideradas -0, 10, 20 y 30 metros-. Asimismo, las variables de salida serían los valores de $P_g(x,N)$ y $P_w(x,N)$ para el mismo conjunto de x en cada capa, después de N segundos de “tiempo simulado”. En cualquier caso, todos estos valores de presión, tanto los de entrada como salida, se expresan en [kPa].

10.3. Visualización y control

Las vistas y controles de la simulación implementada para este modelo han presentado una serie de características distintivas dados ciertos requisitos hallados. Nuevamente ha sido necesaria la implementación de vistas adicionales para poder satisfacer todas las necesidades de control y visualización sin dificultar la ejecución en entornos de baja resolución. Además, dada la velocidad de simulación habitual del software creado mediante EJS, la resolución base de segundo introducida por la formulación del algoritmo y los grandes periodos de tiempo a simular, ha sido necesaria la introducción de mecanismos de control especiales.

La vista principal, *Gas & Water Pressure in Mining Drainage* (figura 40), dispone de la mayoría de elementos de control en la columna izquierda. Así, de arriba hacia abajo, se halla en primer lugar los controles de inicio/pausado y reinicio de la simulación. Bajo estos botones se sitúan los casilleros seleccionables para escoger que vistas adicionales han de ser visibles. Debajo se muestra el tiempo transcurrido de simulación, en escala de días, horas y segundos, y los casilleros para escoger que capas de profundidad queremos ver reflejadas en la vista principal. Así, para cada capa seleccionada, veremos en forma de gráfica como la presión de agua y gas varía al aumentar la distancia del agujero de drenaje. Por último se incluye un campo de valor numérico editable y un casillero seleccionable de uso especial.

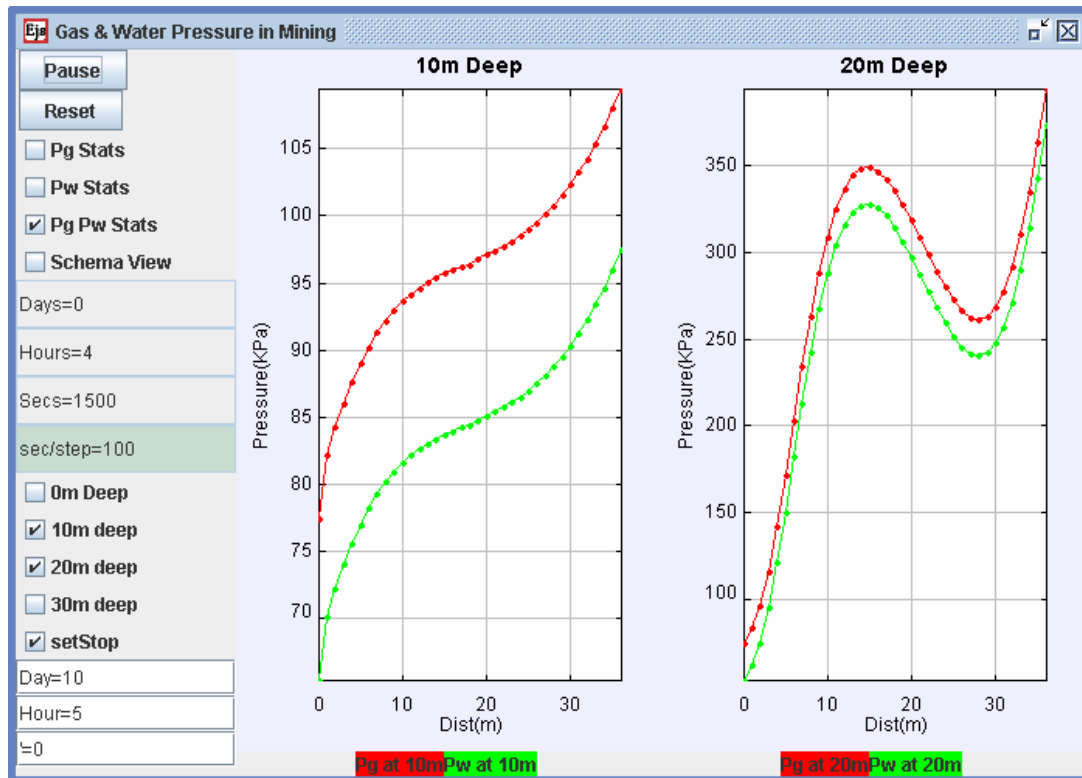


Fig. 40: Vista principal con presiones a 10m y 20m de profundidad, con parada en 10 días 5horas

El primer de estos controles especiales, el campo numérico editable, permite definir el número de segundos transcurridos en cada iteración de simulación. Esto permite, a costa de la precisión, aumentar el tiempo de simulación abarcable en el mismo periodo de tiempo real. Aún así, como ya se ha mencionado, se desaconseja editar este valor, dado que propicia que el algoritmo se vuelva inestable haciendo inútil la simulación. El segundo de estos controles, el casillero *setStop*, permite fijar un punto temporal de interés. De este modo, al marcarlo como verdadero, aparecen tres campos numéricos que permiten introducir un valor de tiempo en días, horas y/o segundos. Una vez transcurrido un periodo de tiempo de simulación igual al indicado, la simulación se pausará automáticamente. Así es posible ver el estado de la simulación, para un momento dado, incluso aunque se tarden horas en alcanzarlo, sin depositar constante atención en la simulación.

Las vistas seleccionables como *Pg Stats* y *Pw Stats* (figura 41) permiten ver las presiones correspondientes al gas y al agua respectivamente. Estas vistas ofrecen precisa información numérica de los puntos de interés principales en todas las capas de profundidad estudiadas, dando los valores de presión en [kPa] tal y como se hace en el resto del modelo.

Ejs Gas Pressure							
Draining	at 0m	at 4m	at 6m	at 9m	at 16m	at 25m	at 35m
00m into	39.702	73.698	77.965	79.742	80.000	80.000	80.000
10m into	42.061	80.704	87.236	91.877	95.856	99.501	108.358
20m into	79.201	174.672	204.225	238.281	287.721	328.611	387.907
30m into	207.780	453.546	528.997	616.704	749.710	836.325	880.365

Ejs Water Pressure							
Draining	at 0m	at 4m	at 6m	at 9m	at 16m	at 25m	at 35m
00m into	27.690	61.698	65.965	67.742	68.000	68.000	68.000
10m into	30.049	68.704	75.236	79.876	83.856	87.502	96.358
20m into	58.197	153.672	183.225	217.281	266.721	307.611	366.907
30m into	195.689	325.446	400.898	487.704	618.011	702.526	747.315

Fig. 41: Vistas con presiones numéricas de agua y gas respectivamente

La última vista (figura 42).con información de carácter científico es seleccionable como *Pg & Pw Stats* en la vista principal, seleccionando *Gas & Water Pressure*. Esta vista ofrece la posibilidad de ver gráficamente todos los valores de presión en cada capa, agrupados según se refieran a la presión de agua o gas. Estas gráficas, a diferencia de las mostradas en la vista principal, permiten ver todas las presiones relativas al gas o al agua en una sola gráfica, ofreciendo una percepción más clara de cómo la presión varía con la profundidad.

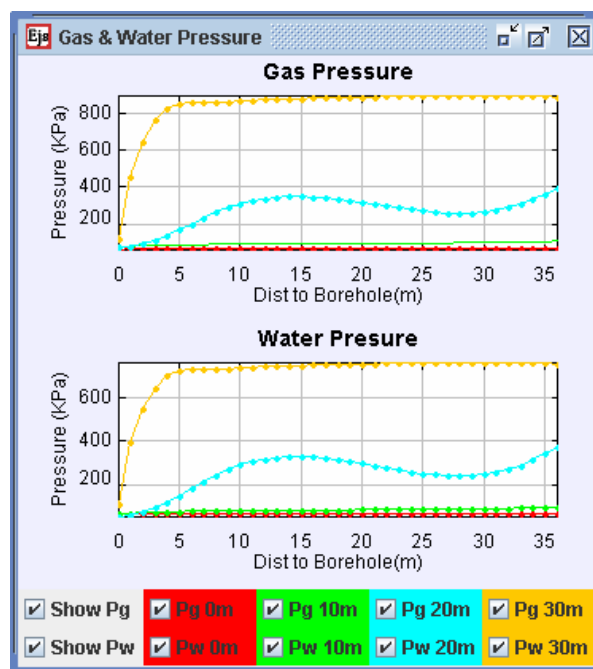


Fig. 42: Vistas resumidas de agua y gas

Finalmente se ofrece una representación más visual de las diferencias y los cambios de presión. De esta manera, en cada punto de interés de cada capa, se aprecia como evoluciona la presión, y por tanto el flujo hacia el drenaje, en relación a los valores de presión iniciales de cada punto.

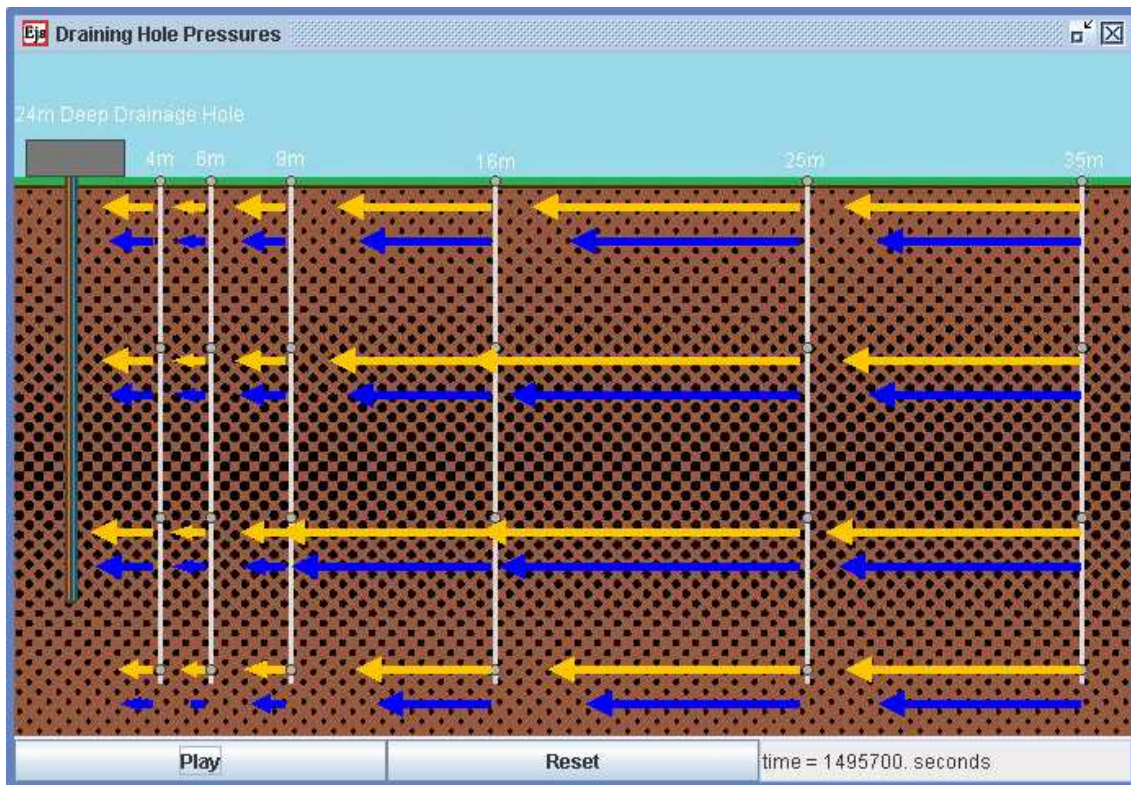


Fig. 43: Esquema visual de la evolución de las presiones

11. Modelado del flujo de metano en los Everglades de Florida

11.1. Introducción

Uno de los mayores riesgos a los que se enfrenta actualmente el planeta Tierra como ecosistema global es el excesivo efecto invernadero. El efecto invernadero es un fenómeno por el cual diversos gases atmosféricos retienen parte de la energía que emite el suelo por haber sido calentado por la radiación solar. Este proceso natural ha permitido regular la temperatura terrestre, evitando el excesivo enfriamiento, y permitiendo florecer la vida. No obstante, la actividad humana lo ha intensificado enormemente con las emisiones de gases, convirtiéndolo en un riesgo.

Uno de los considerados como principales gases de efecto invernadero es el metano. Y mientras que conjuntamente, las emisiones derivadas de las distintas actividades humanas son mayores, la mayor fuente individual de metano atmosférico es el propio medio ambiente: los humedales. Dentro de los humedales se enmarcan los Everglades de Florida, una gran región pantanosa de clima subtropical en Estados Unidos.

El estudio de las emisiones de metano naturales puede ayudar a comprender la naturaleza de efecto invernadero, y como sería éste sin interferencia humana. Éste sería un buen punto de partida para investigar como combatir el calentamiento global de forma efectiva, minimizando el impacto sobre la actividad humana.

11.2. Descripción del modelo matemático

El modelo estudia el flujo de metano en los Everglades de Florida a través de sus distintos estratos, considerando otras variables y factores ambientales [JAM93]. Específicamente, se simula el comportamiento de:

- 1. Acetato (X_1)**
- 2. Sulfato (X_2)**
- 3. Bacterias metanógenas (X_3)**
- 4. Bacterias de sulfato (X_4)**
- 5. Metano (X_5)**

- 6. Bacterias oxidantes de metano (X_6)
- 7. Oxígeno (X_7)

Para ello, el modelo describe cuatro capas explícitas en los Everglades: *agua suprayacente (IV)*, *materia algal (III)*, *sedimento superficial activo(II)*, *sedimento profundo(I)*; y siete compartimentos: acetato (X_1), sulfato (X_2), bacterias metanógenas (X_3), bacterias de sulfato (X_4), metano (X_5), bacterias oxidantes de metano (X_6) y oxígeno (X_7). En las capas de sedimento están presentes todos los compartimentos, mientras que las bacterias anaeróbicas y microaerófilas (X_3 , X_4 y X_6) son excluidas de las capas de aguas (ver figura 44). El modelo también describe cuatro tipos de flujos de materia: flujos de metano y oxígeno a la *atmósfera (V)*, flujos difusivos de metano, oxígeno, sulfato y acetato entre capas ($FLUX_{i(j,k)}$), entradas de fuentes externas, y flujos de un compartimento hacia otro dentro de la misma capa. Todas estas características se implementan a través de un sistema de ecuaciones diferenciales ordinarias, con una serie de funciones para simular el comportamiento de varios factores ambientales.

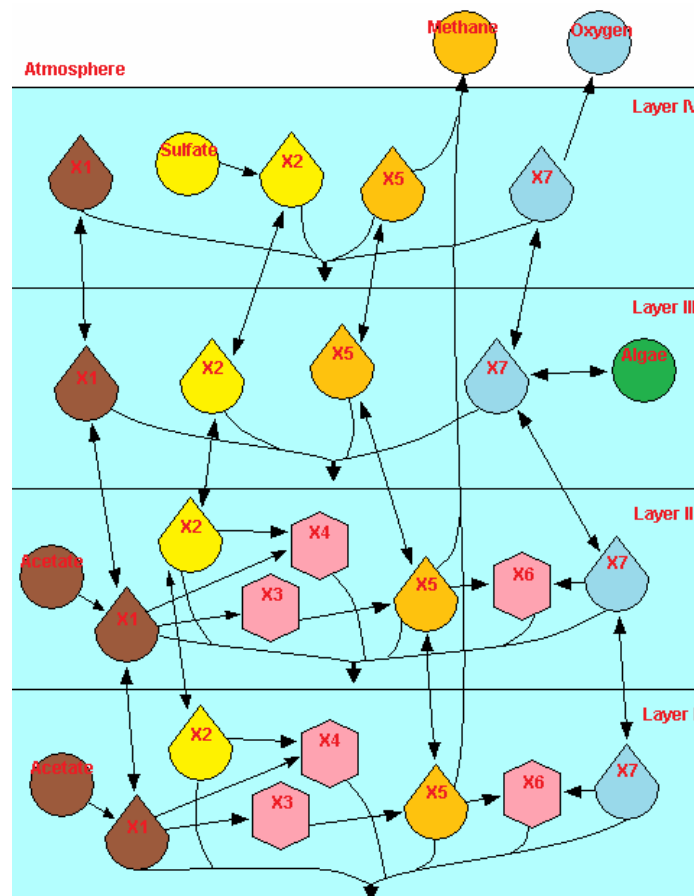


Fig. 44: Esquema de flujos del modelo de metano en los Everglades

Esto hace posible el estudio de la evolución del acetato (X_1), cuya producción y entrada al sistema (F_{01}) tiene lugar en las *capas de sedimento (I y II)*. Las bacteria metanógenas y las del sulfato -(X_3) y (X_4) respectivamente- compiten por este acetato. Para poder emplear el acetato las bacterias del sulfato también necesitan de éste último, el cual es suministrado en el *agua suprayacente (IV) (F_{02})*. Las bacterias metanógenas (X_3) producen metano (X_5), el cual es consumido por las bacterias oxidantes (X_6) del metano o es retirado del sistema a través de las plantas o por difusión a través de las capas y es liberado a la *atmósfera (V)*. El oxígeno (X_7) es empleado por los oxidadores de metano para oxidar el metano, entra en el sistema disolviéndose en el *agua suprayacente (IV)*, y es producido o consumido por las algas de la capa de materia algal (*III*) (F_{07}).

El sistema de EDO que describe el modelo simula los flujos entre capas y compartimentos. Cada compartimento es representado por una X_i , habiendo distintas instancias para cada capa. Los flujos entre capas de un mismo compartimento se describen mediante una ecuación general de flujo $FLUX_{i(j,k)}$. Esta ecuación multiplica una tasa de difusión por la diferencia en las concentraciones del compartimento i en las capas adyacentes j y k . Esto se divide por la distancia entre los centros de cada capa, y se multiplica por el área entre las capas. Los números entre paréntesis, que representan las capas adyacentes, están dispuestos en orden de donante a receptor. Los flujos para una capa j pueden ser positivos -flujo de k a j - o negativos -el flujo es de j a k -. Otras ecuaciones del sistema están modeladas de forma parecida, como $FLUX_{50IV}$ y $FLUX_{07IV}$, aunque en este caso se ignora los volúmenes de las capas, ya que el volumen de la atmósfera no se podría definir.

Acetate(X_1)

$$IV.\delta X_1 = -FLUX_{1(IV,III)} - F_{10}$$

$$III.\delta X_1 = FLUX_{1(IV,III)} - FLUX_{1(III,II)} - F_{10}$$

$$II.\delta X_1 = F_{01(II,T)} + FLUX_{1(III,II)} - FLUX_{1(II,I)} - F_{13} - F_{14} - F_{10}$$

$$I.\delta X_1 = F_{01(I,T)} + FLUX_{1(II,I)} - F_{13} - F_{14} - F_{10}$$

where

$$FLUX_{L(j,k)} = D_L \times AREA \times (X_{L(j)}/VOL_j - X_{L(k)}/VOL_k) / DIST_{jk}$$

$$DIST_{jk} = (DEPTH_j + DEPTH_k) / 2$$

$$F_{10} = X_1 \times \lambda_1$$

$$F_{13} = X_3 \times \tau_{13} \times (X_1 / (K_{13} + X_1)) \times \left(+ [1 - X_7 / (X_7 + K_{73})] \right)$$

$$F_{14} = X_4 \times \tau_{14} \times (X_1 / (K_{14} + X_1)) \times \left(+ [1 - X_7 / (X_7 + K_{74})] \right)$$

Sulfate(X_2)

$$IV.\delta X_2 = F_{2(IV,T)} - FLUX_{2(IV,III)} - F_{20}$$

$$III.\delta X_2 = FLUX_{2(IV,III)} - FLUX_{2(III,II)} - F_{20}$$

$$II.\delta X_2 = FLUX_{2(III,II)} - FLUX_{2(II,I)} - F_{24} - F_{20}$$

$$I.\delta X_2 = FLUX_{2(II,I)} - F_{24} - F_{20}$$

where

$$F_{20} = X_2 \times \lambda_2$$

$$F_{24} = X_4 \times \tau_{24} \times (X_2 / (K_{24} + X_2)) \times \left(\left[1 - X_7 / (X_7 + K_{74}) \right] \right)$$

Methanogens(X_3)

$$II.\delta X_3 = F_{13} \times Y_3 - F_{30}$$

$$I.\delta X_3 = F_{13} \times Y_3 - F_{30}$$

where

$$F_{30} = X_3 \times \lambda_3$$

SulfateBacteria(X_4)

$$II.\delta X_4 = F_{14} \times Y_4 - F_{40}$$

$$I.\delta X_4 = F_{14} \times Y_4 - F_{40}$$

where

$$F_{40} = X_4 \times \lambda_4$$

Methane(X_5)

$$IV.\delta X_5 = -FLUX_{50(I,V)} - FLUX_{50(IV,III)} - F_{50}$$

$$III.\delta X_5 = FLUX_{50(IV,III)} - FLUX_{50(III,II)} - F_{50}$$

$$II.\delta X_5 = FLUX_{50(III,II)} - FLUX_{50(II,I)} - F_{35} - F_{56} - F_{50} - PFLX_{50(II)}$$

$$I.\delta X_5 = FLUX_{50(II,I)} - F_{35} - F_{56} - F_{50} - PFLX_{50(I)}$$

where

$$FLUX_{50(IV)} = KL_5 / DEPTH / 2 \times AREA \times (X_{5(IV)} - X_{5sat})$$

$$X_{5sat} = PPM \times \exp(-16.1962 + 99.1624 \times \ln(100/T) + 27.9015 \times \ln(T/100))$$

T = temperature

$$PFLX_{50(II)} = PFMAX_{(II)} \times X_{5(II)} / (K_{50} + X_{5(II)})$$

$$PFLX_{50(I)} = PFMAX_{(I)} \times X_{5(I)} / (K_{50} + X_{5(I)})$$

$$FLUX_{50Total} = FLUX_{50(IV)} + PFLX_{50(II)} + PFLX_{50(I)}$$

$$F_{35} = F_{13}$$

$$F_{50} = X_5 \times \lambda_5$$

$$F_{56} = X_5 \times \tau_{56} \times (X_5 / (K_{56} + X_5)) \times \left(\left[1 - \left[B_{76} - X_7 / (B_{76} - P_{76}) \right] \right] \right) \times KM$$

MethaneOxydizer(X_6)

$$II.\delta X_6 = F_{56} \times Y_6 - F_{60}$$

$$I.\delta X_6 = F_{56} \times Y_6 - F_{60}$$

where

$$F_{60} = X_6 \times \lambda_6$$

Oxygen(X_7)

$$IV.\delta X_5 = FLUX_{07(I,V)} - FLUX_{7(IV,III)} - F_{70}$$

$$III.\delta X_5 = F_{07(T)} - FLUX_{7(IV,III)} - FLUX_{7(III,II)} - F_{70}$$

$$II.\delta X_5 = FLUX_{7(III,II)} - FLUX_{7(II,I)} - F_{76} - F_{70}$$

$$I.\delta X_5 = FLUX_{5(II,I)} - F_{76} - F_{70}$$

where

$$FLUX_{07(IV)} = KL_7 / DEPTH / 2 \times AREA \times (X_{7sat} - X_{7(IV)})$$

$$X_{7sat} = 14.562 - 0.41022 \times TEMP(T) + 0.007991 \times TEMP^2(T) - 7.7774 \times 10^{-5} \times TEMP^3(T)$$

$$F_{70} = CBOD \times K_{20}^{(T-20)} \times \frac{X_7}{K_{70} + X_7} + F_{10}$$

$$F_{76} = \rho_{75} \times F_{56}$$

La producción de acetato se simula como una entrada en las capas de sedimento (*I* y *II*). La tasa de producción de acetato ($F_{01(II)}$) es estacional, con una mayor producción en verano y un descenso en la estación de invierno. La producción media en si es baja. En el sedimento profundo (*I*) también se produce acetato, aunque la tasa de producción ($F_{01(I)}$) es baja y constante. Los cambios en las concentraciones de acetato en las *aguas suprayacentes* son función de las pérdidas no consuntivas (F_{10}) y los flujos a otras capas. En el *sedimento superficial* (*III*) los cambios en el acetato son una combinación de entradas, flujos entre capas, pérdidas no consuntivas y pérdidas y consumo de las distintas bacterias (F_{13} y F_{14}). En el *sedimento profundo*, se dan los mismos cambios, exceptuando que no hay flujo hacia abajo, ya que la capa de roca subyacente ha sido considerada impermeable.

La presencia de sulfato inhibe la producción del metano ya que las bacterias oxidantes del sulfato superan a las metanógenas para tomar el acetato si hay bastante sulfato accesible. El sulfato se añade al sistema como una pequeña constante (F_{02IV}) en el agua suprayacente. Éste puede ser perdido de forma no consuntiva (F_{20}), tomado por las bacterias oxidantes (F_{24}), o moverse a través de los sedimentos por difusión general.

El modelo emplea la cinética de Michaelis-Meten para definir el consumo de acetato por parte de los metanógenos (F_{13}). Ya que las bacterias metanógenas son anaeróbicas, el modelo emplea una constante de media saturación inversa que asume que la mitad del consumo máximo de acetato ocurre a una concentración de $0.031 \text{ moles}\cdot\text{m}^{-3}$ de oxígeno. Por encima de esta concentración, el consumo de acetato decae significativamente. Una constante de rendimiento, Y_3 , multiplicada por el acetato tomado define el crecimiento bacteriano. Esta producción bruta menos una tasa de mortalidad no predatoria (F_{30}) da como resultado el crecimiento neto de las bacterias.

Las bacterias del sulfato compiten directamente con las metanógenas por el acetato. El modelo también emplea cinéticas de consumo de Michaelis-Meten para definir el crecimiento de los reductores de sulfato. El sulfato y el acetato se toman en una base 1:1 -mol por mol-. De forma similar a los metanógenos, los reductores de sulfato son anaeróbicos, y están sometidos también a una constante de media saturación inversa. La constante de rendimiento Y_4 multiplicada por el consumo de acetato F_{14} determina la producción bruta de reductores de sulfato. A este valor se le resta la mortalidad no predatoria F_{40} para obtener la producción neta.

El flujo de metano hacia la atmósfera se calcula sumando los flujos de las distintas vías que puede seguir. Desde las aguas poco profundas la mayoría del flujo transcurre a través de plantas. La difusión habitual desplaza también el metano de los sedimentos hacia el agua suprayacente (IV). El movimiento de metano entre el agua suprayacente y la atmósfera es modelado empleado un coeficiente de intercambio de fase líquida (KL_5). Para simular el intercambio con la atmósfera este coeficiente se multiplica por la diferencia entre el metano en el agua suprayacente y la solubilidad del metano [URL09]. La producción de metano es equivalente a la toma de acetato en base 1:1 -mol por mol-. Los cambios en las cantidades de metano son combinación del intercambio de fase líquida entre el agua suprayacente y la atmósfera ($FLUX_{50IV}$), flujos entre las capas, el consumo por parte de los oxidantes de metano (F_{56}) y las pérdidas no consuntivas (F_{50}).

La oxidación del metano puede tener un impacto en el flujo de metano hacia la atmósfera. El consumo del metano por sus oxidantes se modela de forma similar a la toma de acetato por los metanógenos. Dado que los oxidantes de metano son microaerófilos se emplea un coeficiente KM que indicará la razón del consumo que realizarán. El oxígeno también es consumido por los oxidantes de metano a razón de 3.2 moles de O_2 por cada mol de CH_4 . Este consumo bruto es multiplicado por una constante de rendimiento Y_6 , dando la producción bruta de oxidantes de metano. A

esta cantidad se le restan las pérdidas no consuntivas (F_{60}), obteniendo el crecimiento neto.

El flujo de oxígeno a la atmósfera se modela de forma parecida al de metano hacia la misma. Una vez el modelo determina la solubilidad del oxígeno, la diferencia entre la saturación y la cantidad de oxígeno presente en el agua suprayacente se multiplica por un coeficiente de aireación (KL_7) para determina el flujo de entrada o salida de oxígeno. La materia algal produce oxígeno durante ciertas porciones del año, y lo consume durante el resto. El modelo emplea una constante de *demanda carbonosa de oxígeno* (**CBOD**) para determinar el oxígeno retirado por la descomposición de la materia orgánica. Esta constante se multiplica por una tasa de consumo de oxígeno (K_{20}), que es afectada por la temperatura con un coeficiente de temperatura, y por el oxígeno con una cinética de Michaelis-Menten. Los cambios en el compartimento del oxígeno están modelados como aireación ($FLUX_{07IV}$), el flujo entre capas, la entrada o salida en la capa de materia algal, el consumo por descomposición de materia orgánica *CBOD* (F_{70}), el consumo de los oxidantes de metano (F_{76}) y el debido a la pérdida de acetato (F_{10}).

Como se ha podido apreciar, el modelo emplea varias docenas de parámetros además de varias entradas. A continuación se listan todos los parámetros agrupado por el compartimento para cuyo estudio son relevantes principalmente:

Acetato (X_1)

- Producción de acetato en *sedimento superficial (II)*, $F_{01(II)} = -0.03$ a 0.09 [mol m⁻² día⁻¹]
- Producción de acetato en *sedimento profundo (I)*, $F_{01(I)} = 0.01$ [mol m⁻² día⁻¹]
- Pérdidas no consuntivas, $\lambda_1 = 0.01$ [mol mol⁻¹ día⁻¹]
- Rango intermedio para sedimento de lago, $D_1 = 1.3$ [cm² día⁻¹]

Sulfato (X_2)

- Sulfato añadido a *aguas suprayacentes (IV)*, $F_{02(IV)} = 0.2$ [mol día⁻¹]
- Pérdidas no consuntivas, $\lambda_2 = 0.01$ [mol mol⁻¹ día⁻¹]
- Rango intermedio para sedimento de lago, $D_2 = 1.3$ [cm² día⁻¹]

Metanógenos (X_3)

- Consumo máximo de acetato por metanógenos, $\tau_{13} = 0.0157$ [moles acetato g de célula⁻¹ día⁻¹]
- Constante de media saturación, $K_{13} = 0.05$ [mol m⁻³ día⁻¹]
- Constante de rendimiento, $Y_{13} = 2.136$ [g de células mol⁻¹]
- Mortalidad no predatoria, $\lambda_3 = 0.01$ [g g⁻¹ día⁻¹]
- Tolerancia al oxígeno para media saturación de metanógenos, $K_{73} = 0.031$ [mol oxígeno m⁻³]

Reductores de sulfato (X_4)

- Consumo máximo de acetato por reductores de sulfato, $\tau_{14} = 0.01$ [moles acetato g de célula⁻¹ día⁻¹]
- Consumo máximo de sulfato por reductores de sulfato, $\tau_{24} = 0.01$ [moles sulfato g de célula⁻¹ día⁻¹]
- Constante de media saturación, $K_{14} = 0.05$ [mol m⁻³ día⁻¹]
- Constante de rendimiento, $Y_{14} = 2.136$ [g de células mol⁻¹]
- Mortalidad no predatoria, $\lambda_4 = 0.01$ [g g⁻¹ día⁻¹]
- Constante de media saturación para sulfato, $K_{24} = 0.068$ [mol m⁻³]
- Tolerancia al oxígeno para media saturación de reductores de sulfato, $K_{74} = 0.031$ [moles oxígeno m⁻³]

Metano (X_5)

- Concentración media de metano en la atmósfera, $PPM = 0.0654$ [mM]
- Tasa de difusión, $D_5 = 5.0$ [cm² día⁻¹]
- Coeficiente de intercambio de fase líquida, $KL_5 = 0.24$ [m día]
- Flujo de metano emitido por plantas, $PFMAX_{50(l)} = 0.004$ [mol m⁻³ día⁻¹]
- Flujo de metano emitido por plantas, $PFMAX_{50(l)} = 0.001$ [mol m⁻³ día⁻¹]
- Constante de media saturación, $K_{50} = 0.4$ [mol m⁻³ día]
- Perdidas no consuntivas, $\lambda_5 = 0.01$ [mol mol⁻¹ día⁻¹]

Oxidantes de metano (X_6)

- Consumo de metano, $\tau_{56} = 0.211$ [moles CH₄ g⁻¹ de células día⁻¹]

- Constante de rendimiento, $Y_6 = 2.97$ [g peso seco mole⁻¹ CH₄]
- Constante de media saturación, $K_{56} = 0.141$ [moles CH₄ m⁻³]
- Inhibición total por oxígeno, $K_{76} = 0.125$ [moles O₂ m⁻³]
- Oxígeno mínimo requerido, $P_{76} = 0.003$ [moles O₂ m⁻³]
- Concentración óptima de oxígeno, $B_{76} = 0.031$ [moles O₂ m⁻³]
- Mortalidad no predatoria, $\lambda_6 = 0.01$ [g g⁻¹ día⁻¹]
- Razón de oxígeno consumido por mol de CH₄ producido, $\rho_{75} = 3.2$ [moles O₂ moles⁻¹ CH₄]

Oxígeno (X7)

- Demanda de oxígeno para descomposición de sedimentos, $CBOD = 2$ [g C m⁻³]
- Coeficiente de temperatura, $\theta_7 = 1.047$ [sin unidades]
- Tasa de descomposición a 20 [°C], $K_{20} = 0.0031$ [moles O₂ g⁻¹ C día⁻¹]
- Tasa de difusión del oxígeno, $D_7 = 1.3$ [cm² día⁻¹]
- Coeficiente de intercambio de fase líquida para el oxígeno, $DL_7 = 2.0$ [m día⁻¹]
- Constante de media saturación, $K_{70} = 0.7$ [mol m⁻³ día⁻¹]

Los distintos compartimentos X_i se inicializan con los siguientes valores dados para cada capa (tabla 4):

Layer	Depth (m)	Acetate (moles/m ³)	Sulfate (moles/m ³)	Methanogens (g/m ³)	Sulfate Bacteria (g/m ³)	Methane (moles/m ³)	Methane Oxidizers (g/m ³)	Oxygen (moles/m ³)
Overlying Water	0.10	0.0	0.1	0.0	0.0	0.0	0.0	0.2188
Algal mat	0.05	0.0	0.1	0.0	0.0	0.0	0.0	0.2188
Upper Sediment	0.05	0.02	0.03	10.0	0.5	0.0	10.0	0.0313
Lower Sediment	0.2	0.02	0.003	10.0	0.5	0.0	0.5	0.0001

Tabla 4: Concentraciones iniciales de los compartimentos

Con lo cual, el modelo puede ser representado por el siguiente diagrama de bloques (figura 45):

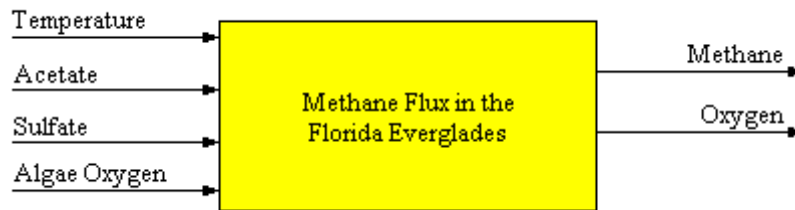


Fig. 45: Diagrama de bloques del modelo de metano en los Everglades

Donde se observan como entradas al sistema las siguientes variables:

- Temperatura, T [°C]
- Producción de acetato, $F_{01(I)}$ y $F_{01(II)}$ [$\text{mol}\cdot\text{m}^{-2}\text{ día}^{-1}$]
- Producción de sulfato, $F_{02(IV)}$ [$\text{mol}\cdot\text{m}^{-2}\text{ día}^{-1}$]
- Oxígeno producido por algas, $F_{07(T)}$ [$\text{mol}\cdot\text{m}^{-2}\text{ día}^{-1}$]

Y como variables de salida estarían presentes:

- Flujo de metano a la atmósfera, $FLUX_{50total}$ [$\text{mol}\cdot\text{m}^{-2}\text{ día}^{-1}$]
- Intercambio de oxígeno con la atmósfera, $F_{07(IV)}$ [$\text{mol}\cdot\text{m}^{-2}\text{ día}^{-1}$]

11.3. Aspectos técnicos

La implementación de este modelo, así como de las vistas para visualización y control de la simulación, ha resultado de una dificultad y complejidad importante. Solamente citar la presencia de 22 variables de estado y 4 variables de entrada descritas con funciones, para calcular 2 variables de salida dan una idea aproximada de la complejidad. Además de la ingente cantidad de información manejada por el modelo en ejecución y la complejidad de éste mismo, algunas características de EJS habitualmente deseables han constituido un escollo a salvar.

Un detalle que habitualmente tiene poco peso a la hora de plantear el diseño, el paso de simulación empleado, ha resultado de una importancia e influencia críticas. Así, el artículo describiendo el modelo citaba el modelo con doble paso de simulación: uno

pequeño de una centésima de día, y uno grande de décima de día. Con el fin de obtener una mayor velocidad de simulación -el objetivo es simular periodos de varios años- se planteó inicialmente el uso del paso de simulación menor. No obstante, los resultados obtenidos en las primeras etapas de testeo se mostraron erróneos, dada la inestabilidad del sistema. Esto llevó a cambiar el intervalo a una centésima de día.

Este cambio acrecentó un problema ya detectado. Como se ha mencionado en el capítulo de EJS, cuando se emplea un elemento *Traza*, por defecto guarda el valor de las coordenadas *X* e *Y* de cada punto añadido en cada paso de simulación. Esto supone para cada variable de tipo *double* guardar 36500 elementos por año. En conjunto, para poder obtener información completa sobre 22 variables de estado y las 6 de entrada/salida, hablamos de mantener el equivalente a más de un millón de elementos de tipo *double* en memoria, algo totalmente inviable. La primera alternativa es emplear la opción de los elementos *traza* para limitar la cantidad de puntos guardados. No obstante, aunque pusiéramos un límite realmente elevado, del orden de 10,000 (diez mil), con 100 iteraciones por día de simulación, éste nos permitiría mostrar en las gráficas 100 días, menos de un tercio de año. Mostrar un tercio de año en las gráficas de un modelo cuyo objetivo es simular un proceso para ver como evoluciona a lo largo de años, es inaceptable.

Finalmente, tras buscar una solución al problema, se halló la solución planteando la pregunta adecuada: ¿Es realmente necesario guardar 100 puntos por día para una *traza* de simulación que representará a varios años en un espacio de menos de 250 píxeles? La evidente respuesta negativa dio lugar a la idea ya aplicada en otras visualizaciones de simulación: las vistas no necesitan mostrar todos los datos ni estado por los que pasan las variables de estado ni el sistema en ejecución.

Aplicando este principio, se creó un conjunto de variables duplicando toda variable que fuese a estar presente en alguna *traza* de las distintas vistas. Estas variables duplicadas se actualizan con valores de las variables originales cada 100 iteraciones de simulación mediante código insertado en las *Relaciones Fijas*. Activando la opción de *No Repetir* de las *trazas*, solo se añaden puntos a éstas cuando varían los valores, es decir, cada 100 iteraciones, o una vez por día simulado. De esta manera, se puede ofrecer información sobre la evolución de todas las variables a lo largo de toda la ejecución en forma de gráficas, empleando un PC convencional sin riesgos de fallos o caídas de *Java* o del sistema.

11.4. Control y visualización

La implementación de las vistas para visualización y control de la simulación de este modelo ha resultado de una dificultad y complejidad importante. Tal y como ya se ha comentado, EJS no ofrecía ningún elemento realmente adecuado a nuestra necesidades, y a sido preciso adaptar la salida de datos al funcionamiento del elemento *Traza*. Aún con este escollo salvado, ha sido necesario un notable esfuerzo de diseño para poder implementar un conjunto de vistas completo y preciso, pero no demasiado complejo, y apto para sistemas de baja resolución.

La vista principal (figura 46), como en otros casos, contiene la totalidad de los elementos de control y parte de los elementos de visualización. Éstos últimos se sitúan en la parte central y derecha de la vista. La parte central la ocupan las gráficas de las variables de entrada no constantes. La parte izquierda contiene las gráficas correspondientes a las variables de salida. Esta distribución atípica, con las gráficas de más interés en el lateral en lugar del centro es debida a las políticas de distribución de EJS y el reducido espacio disponible para mantener la compatibilidad con entornos de baja resolución. Por ello las gráficas de variables de salida se han dispuesto al lateral, donde no se ven afectadas por el redimensionamiento al encoger el marco principal.

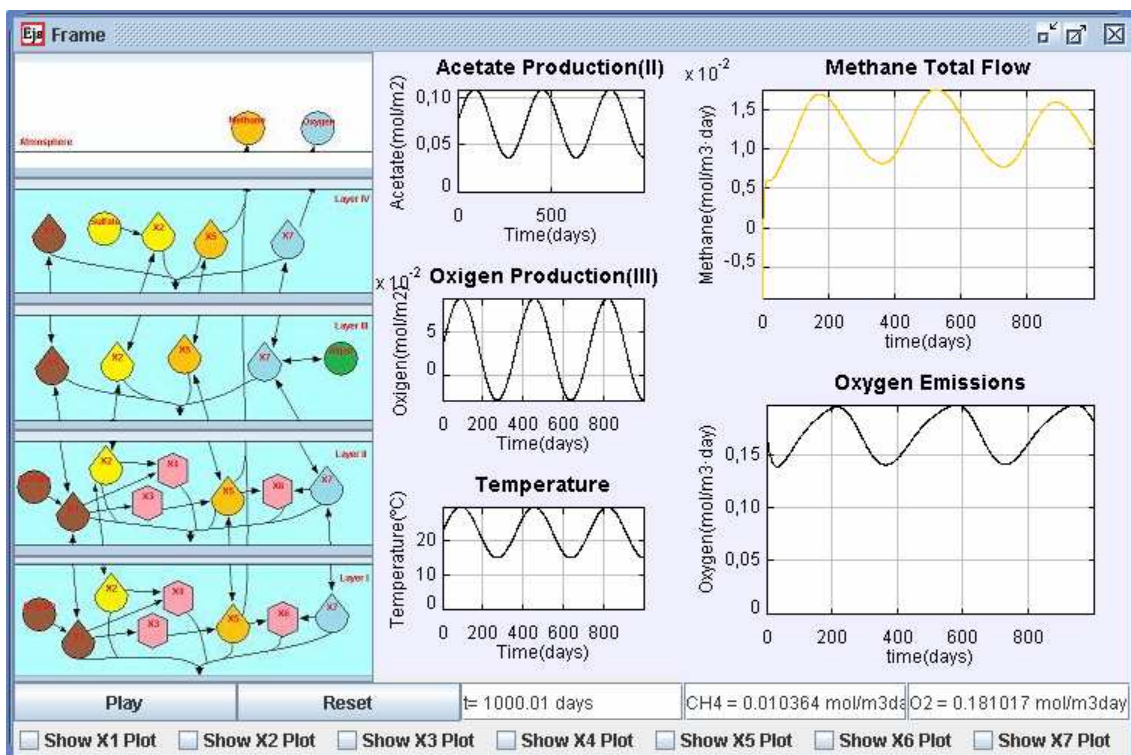


Fig. 46: Vista principal del modelo, tras mil días de simulación

Los elementos de control se distribuyen por el margen inferior e izquierdo de la vista principal. En la parte izquierda se disponen 5 botones en columna, cuyo conjunto forma una representación en miniatura del esquema general de flujos del sistema simulado. Cada botón se corresponde a una capa, y activa la visualización de una vista ampliada del esquema de esa capa, como se detalla más adelante. La parte inferior dispone en línea los elementos de control de ejecución y elementos de tipo *CampoNumerico* para mostrar la evolución numéricamente del tiempo y de las variables de salida con seis decimales de precisión. Por debajo de estos elementos se alinean siete casilleros seleccionables que permiten activar las vistas con las gráficas para cada compartimento.

Las vistas *Layer_i* (figura 47) muestran la parte del esquema de flujos correspondiente a la capa *i*, pero añadiendo información sobre las concentraciones de cada compartimento presente en esa capa. Se ha empleado en el esquema un código de formas y colores para diferenciar los distintos elementos, bacterias, y si son flujos de entrada o salida, bacterias o cantidades. Para cada flujo o cantidad de un compartimento reflejada, se da un real con cinco decimales de precisión. En un sistema de resolución media-alta es enteramente posible disponerlas como una columna, mostrando el esquema de flujos del sistema, y ver como evolucionan los distintos compartimentos paralelamente.

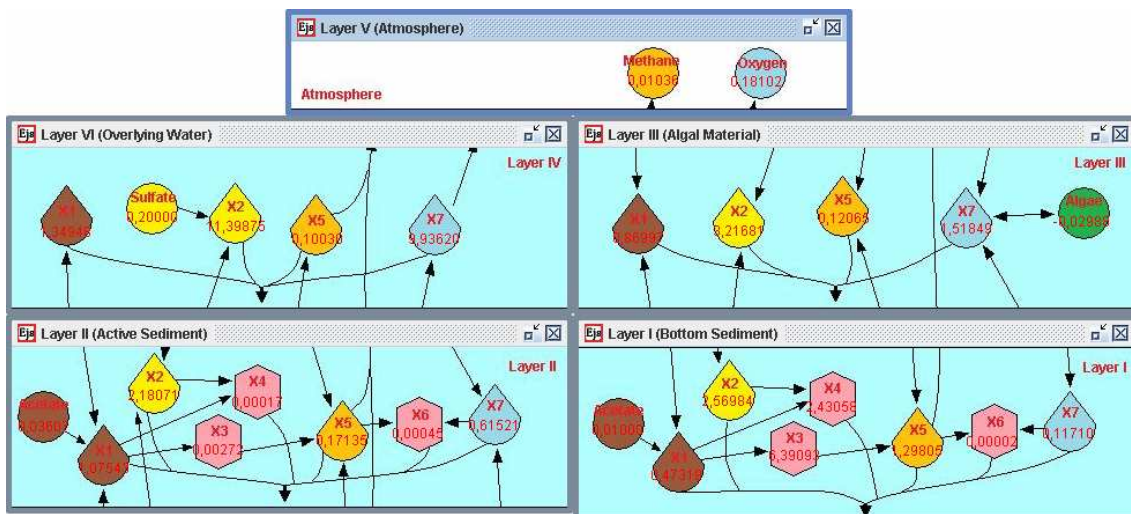


Fig. 47: Vistas de capas expandidas

El último conjunto de vistas adicionales, como ya se ha mencionado, muestra gráficas detalladas para cada compartimento, reflejando en la misma gráfica la evolución de un mismo compartimento en todas sus capas (figura 48). Esta distribución permite ver la diferencia en las cantidades de un mismo compartimento a través de las capas,

dándonos una idea de hasta que punto cada compartimento puede propagarse y distribuirse a través de éstas. A lo largo de las distintas vistas se ha empleado la misma codificación de colores para las trazas de cada.

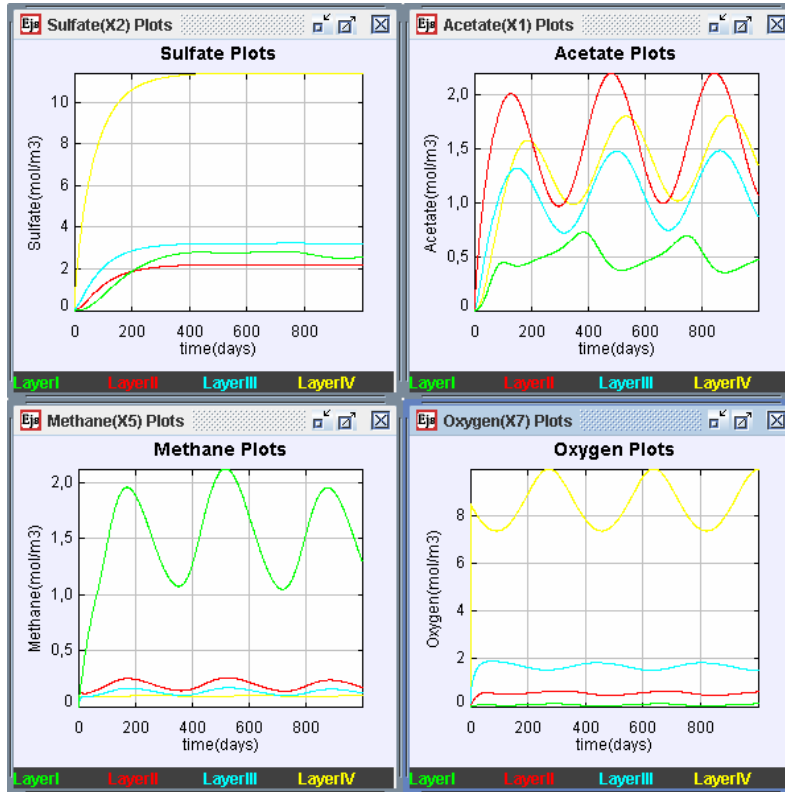


Fig. 48: Gráficas de compartimento, algunos ejemplos

12. Documentación y distribución de las simulaciones

12.1. Documentación de las simulaciones

Como se ha comentado en otros apartados del proyecto, la finalidad de éste es principalmente educativa, ya que algunas de las simulaciones pasarán a formar parte del material de la asignatura SSD –Simulación de Sistemas Dinámicos-, de la ETSEIB, en un futuro no muy lejano. Pero también se desea obtener una dimensión más divulgativa, haciendo que puedan acceder a las simulaciones la mayor cantidad de personas posible. Para cualquiera de los fines es preciso que las simulaciones contengan la información necesaria para su comprensión y contextualización. Esta labor de documentación digital se realiza en EJS, cuyo pequeño editor de HTML permite crear texto formateado e introducir imágenes sin casi tener conocimientos de HTML.

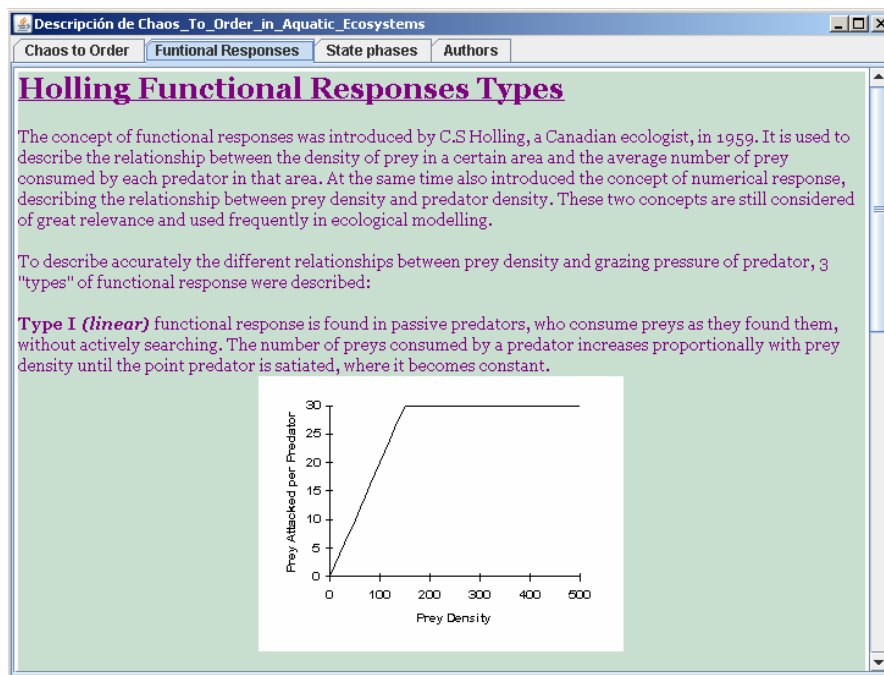


Fig. 49: Ejemplo de páginas de descripción de una simulación como ejecutable independiente

De esta manera se han documentado completamente las simulaciones, ofreciendo explicaciones sobre la finalidad y contexto de cada sistema modelado, descripciones detalladas de los modelos matemáticos, listados de ecuaciones, variables y parámetros, e información relativa a conceptos presentes en los modelos de especial interés. Toda esta información será visible de una manera u otra en función del

método de distribución escogido. Si por ejemplo se empaqueta una simulación como un único ejecutable de *Java* –extensión *.jar*-, al ejecutar el *applet* como una aplicación, además de sus vistas, se mostrará una vista adicional, la cual contendrá la información dispuesta en las páginas creadas con el editor de HTML de EJS (figura 49).

No obstante, dado que además de satisfacer unos objetivos educativos, también se pretende dar mayor difusión a los modelos, se ha optado por una distribución alternativa en forma de páginas web. Así, EJS generará la simulación incrustada en un conjunto de páginas web, que también contendrán la documentación descrita (figura 50).

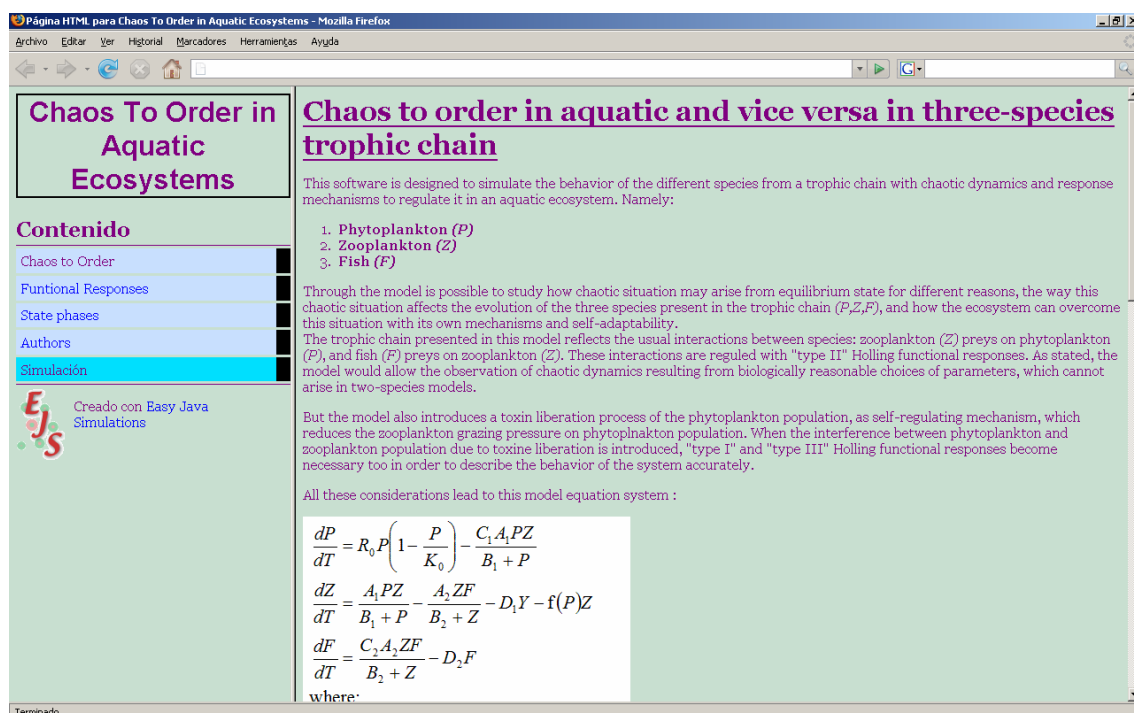


Fig. 50: Ejemplo de página HTML de documentación de una simulación.

12.2. Distribución de las simulaciones

EJS ofrece una serie de opciones automatizadas para la distribución de las simulaciones creadas. Tal y como se ha comentado, se han realizado una serie de páginas HTML, conteniendo tanto las simulaciones, como las descripciones implementadas como documentación del modelo. Este proceso lo puede realizar EJS de forma automatizada, simplificando enormemente la tarea de distribución de las simulaciones.

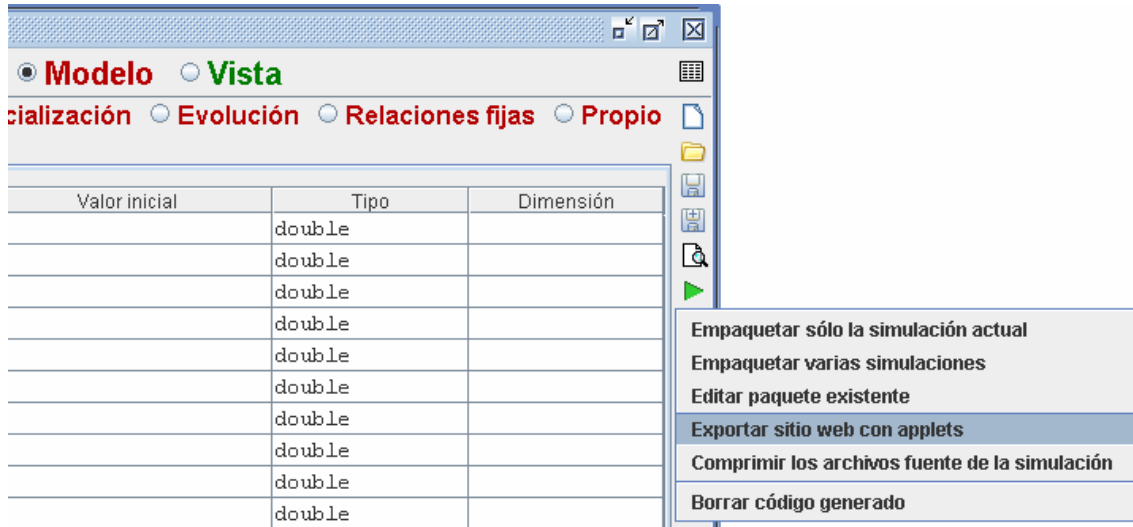


Fig. 51: Detalle de las opciones de distribución de EJS

Empleando estas opciones (figura 51), se exportan las simulaciones como una página web generada automáticamente. Esta página puede contener una única simulación, o un conjunto de simulaciones, creando EJS una página a modo de índice para este grupo de simulaciones. Este índice vincula las páginas necesarias para todas las simulaciones.

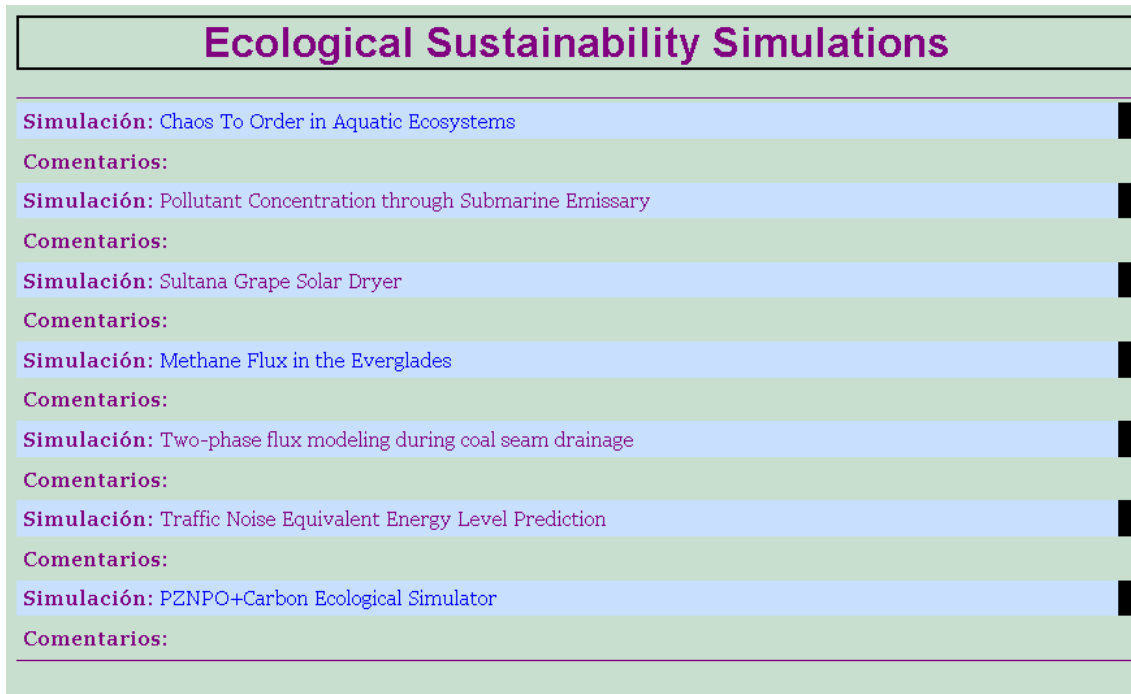


Fig. 52: Página de grupo generada por EJS a modo de índice

Estas páginas se crean en una carpeta, imitando la estructura en la que se hayan dispuesto en el directorio de trabajo –definido por defecto por EJS en la carpeta `./workspace/source-`, conteniendo todo lo necesario para las simulaciones.

Chaos To Order in Aquatic Ecosystems

Contenido

- Chaos to Order
- Functional Responses
- State phases
- Authors
- Simulación

Creado con Easy Java Simulations

La vista de la simulación debe verse justo bajo esta línea:

PZF3D Phytoplankton Zooplankton Fish Plots

$p = 0.749000000$
 $z = 0.161000000$
 $f = 9.920000000$

$a1 = 5,000$
 $a2 = 0,100$
 $b1 = 3,000$
 $b2 = 2,000$
 $d1 = 0,400$
 $d2 = 0,010$
 $ro = 0,000$

Steps = 0

Play Reset
Save Load Show Cam Setting

Puede controlarla usando JavaScript:
Marcha Pausa Reset Paso Despacio Rápido Mas rápido

Fig. 53: Página de ejemplo creada por EJS con la simulación de un modelo del proyecto

Una vez creadas estas páginas (figura 53), con su página de grupo (figura 52), y una carpeta conteniendo todas las imágenes, clips de audio o video, ejecutables, ficheros HTML, el siguiente paso es su despliegue en un servidor web conectado a Internet. Para este proyecto se emplea el servidor `model.upc.edu`, hospedado en equipos del departamento de ESAII. Así, una vez están desplegadas las páginas web en el servidor, éste podrá servir las vía Internet a usuarios interesados.

13. Conclusiones

Finalmente el proyecto ha sido completado satisfactoriamente. Los objetivos han sido cumplidos en su totalidad, teniendo en cuenta los criterios asociados a éstos. Pese a haber sido algo problemático en algunos aspectos –como el diseño de vistas-, se han buscado soluciones fuera de caja –o fuera de *applet* para las vistas-, o se ha buscado un compromiso para satisfacer en cierta medida todos los requisitos.

Las simulaciones han sido implementadas por completo, con vistas ricas y elaboradas, mostrando un comportamiento acorde con lo esperado de los distintos modelos, pese a algunos fallos en documentación empleada como fuente de información. Los requisitos complementarios, uno de los aspectos que podía acabar siendo más difícil de satisfacer, finalmente han sido cumplidos con creces. Las simulaciones son visibles plenamente en navegadores y sistemas en baja resolución sin suponer molestia alguna, incluyendo la práctica totalidad de modelos diversas alternativas de visualización. De este modo es posible mantener el rigor exigible a un PFC, acercando las interfaces de usuario a un público de perfil menos científico, mediante animaciones, representaciones pictóricas e incluso sonido.

En cuanto a la herramienta EJS, se puede decir que ha sido seguramente la elección óptima, estando a la altura de sus competidores funcionalmente pese a ser gratuito. Cabe mencionar que EJS se encuentra aún en desarrollo, habiendo pasado por las versiones 3.46, 3.47 y 4.0 durante la elaboración de este PFC, incluyendo mejoras en cada una de ellas. De hecho, a inicios de 2009 ha salido una nueva versión 4.1, con diversas mejoras interesantes, entre ellas nuevos elementos que podrían haber contribuido a simplificar el diseño visual de las simulaciones. No obstante, también es necesario mencionar como el editor HTML en la versión 4.0 sufre de cierta tendencia a provocar fallos en la aplicación *Java*, siendo aconsejable el uso complementario de un editor de texto plano o de HTML. Detalles como, pese a ser importantes, no hacen desmerecer un entorno de desarrollo en conjunto excelente.

A nivel personal el proyecto ha sido especialmente enriquecedor e interesante, al no estar centrado únicamente en aspectos informáticos aislados de cualquier otro campo. No olvidemos que la informática es una disciplina transversal, y que por muy grandes que sean sus logros individuales, su mayor contribución al desarrollo humano es el poder que brinda a otras disciplinas para expandir sus campos. Este potencial que brinda, unido a las TIC, la vincula estrechamente a la sostenibilidad, ya que siendo el

estudio de ésta un problema multidisciplinar, de grandes proporciones y de ámbito global; la informática, como disciplina complementaria permite unir esfuerzos multidisciplinarios, otorga enormes poderes de tratamiento y gestión de información, y aúna el esfuerzo de gente de todo el globo. Por supuesto también ha sido interesante aprender y profundizar en temas de carácter más científico que no técnico, aunque a veces haya resultado frustrante tener que enfrentarse a lecturas y problemas complejos de disciplinas totalmente ajenas.

Creo que realizar este proyecto también me ha ayudado a desarrollar algo más las capacidades de organización y planificación, y asumir una mayor iniciativa y decisión, sobretodo en aspectos asociados al diseño, tanto a nivel de código y algoritmia, como a nivel de interfaces de usuario y usabilidad. Asimismo, la redacción de textos marcadamente científicos en inglés, pese a haber sido un desafío, ha sido un excelente ejercicio para afianzar –al menos a nivel escrito- el dominio de una lengua extranjera, algo siempre útil.

Como conclusión final, y a título personal, creo poder decir que estoy satisfecho, por haber cumplido con todo el trabajo pese a algunas adversidades, y especialmente por lo enriquecedor del trabajo realizado, personalmente, y como aportación que pueda suponer tanto a nivel académico para el departamento de ESAll, como a un fin divulgativo en pro de una sociedad más responsable y sostenible. También ha cambiado significativamente mi visión de la informática dentro del mundo, a nivel de posibilidades, roles y responsabilidades, aspecto en el que pocas asignaturas obligatorias hacen hincapié.

Bibliografía y Referencias

Modelos y artículos

[BAS88] Basu, A., and Boyd, M.J., "Numerical modelling of two phase flow of gas and water during drainage of a coal seam", *International Journal of Mine Water*, vol. 7, pp. 27-42, 1988.

[DAV07] Davis, M.L., and Cornwell, D.A., *Introduction to Environmental Engineering*, McGraw-Hill, 2007.

[ESP00] Espino Infantes, M., "¿Hasta que punto es predecible el comportamiento de los contaminantes en el mar?", *Jornades "Qui embruta el mar?"*, pp. 159-174, Museu Marítim de Barcelona, Barcelona 6-7 juny 2000.

[JAM93] James, R.T., "Sensitivity analysis of a simulation model of methane flux from the Florida Everglades", *Ecological Modelling*, vol. 68, pp. 119-146, 1993.

[MAN06] Mandal, S., Ray, S., Roy, S., and Jørgensen, S.E., "Order to chaos and vice versa in an aquatic ecosystem", *Ecological Modelling*, vol. 197, pp. 498-504, 2006.

[MES02] Mestres, M., "Three-dimensional simulation of pollutant dispersion in coastal waters", *Ph.D. Dissertation*, Technical University of Catalonia, May 2002.

[SOL05] Solidoro, C., Pastres, R., and Cossarini, G., "Nitrogen and plankton dynamics in the lagoon of Venice", *Ecological Modelling*, vol. 184, pp. 103-124, 2005.

[YAL01] Yaldiz, O., Ertekin, C., and Uzun, H.I., "Mathematical modeling of thin layer solar drying of sultana grapes", *Energy*, vol. 26, pp. 457-465, 2001.

[WU07] Wu, L.C., Wei, C.B., Yang, S.S., Chang, T.H., Pan, H.W., and Chung, Y.C., "Relationship between carbon dioxide/methane emissions and the water quality/sediment characteristics of Taiwan's main rivers", *Air Waste Management Association*, vol. 57, pp. 319-327, Mar 2007.

EJS y alternativas

[ESQ05] Esquembre, F., *Easy Java Simulations The Manual*, Version 3.4, September 2005.

- [URL01] <http://www.um.es/fem/EjsWiki/>
- [URL02] <http://www.mathworks.es/products/matlab/>
- [URL03] <http://www.vensim.com/>
- [URL04] <http://www.iseesystems.com/software/education/StellaSoftware.aspx>

Java y bibliotecas usadas

- [URL05] <http://java.sun.com/javase/6/docs/api/>
- [URL06] <http://commons.apache.org/math/api-1.2/index.html>

Consultas y utilidades varias

- [URL07] <http://es.wikipedia.org>
- [URL08] <http://en.wikipedia.org>
- [URL09] http://www.engineeringtoolbox.com/gases-solubility-water-d_1148.html
- [URL10] <http://classes.entom.wsu.edu/543/Functional.htm>
- [URL11] <http://www.akiti.ca/CubicSpline.html>