



Un laboratorio de matemáticas basado en el estudio de un problema real utilizando el enfoque de los modelos IBM.

Anna Vidal Meló^a, Vicente D. Estruch^b y Francisco J. Boigues^c

Departamento de Matemática Aplicada. Campus de Gandia de la Universitat Politècnica de València.
Grup d'Innovació Educativa i Recerca en Mètries Científiques (GIERMAC)

^aavidal@mat.upv.es, ^bvdestruc@mat.upv.es y ^cfraboipl@mat.upv.es

Abstract

Education in Engineering and Sciences requires math skills oriented to the applications, ie to the mathematical models. In the first degree courses, models usually are introduced from the speech of the teacher, being still difficult for the students to be able for building models using differential equations or difference equations. In this context, the student's work is reduced to experimenting with the given model, and is essential having available a proper assistant for the computations. Individual Based Modeling (IBM) may grant an important role to the student, also in the model design phase. The approach to a problem by IBM models consists in explaining the behavior of a system by the aggregation of the individual behavior of each of the elements of the system. In this work, a practical experience introducing IBM is presented, using as assistants Matlab[®] and / or Octave[®].

Keywords: *Mathematical modeling, IBM, Matlab[®], Octave[®].*

Resumen

La formación del científico o del ingeniero exige competencias matemáticas orientadas hacia las aplicaciones, es decir hacia los modelos matemáticos. En los primeros cursos de grado, los modelos suelen presentarse desde el discurso del profesor, siendo difícil que un alumno sea capaz de construir modelos mediante ecuaciones diferenciales o ecuaciones en diferencias. En este contexto, el trabajo del estudiante se reduce a la experimentación con el modelo dado, y disponer para ello de un asistente para el cálculo adecuado es fundamental. Los modelos basados en el individuo (Individual Based Modeling o IBM), pueden otorgar un mayor protagonismo al alumno, incluso en la fase de diseño del modelo. La aproximación a un problema mediante modelos IBM consiste en explicar el comportamiento de un sistema formado por un número determinado de elementos a partir de la agregación del

Un laboratorio de matemáticas basado en el estudio de un problema real utilizando el enfoque de los modelos IBM

comportamiento individual de cada uno de los elementos del sistema. En este trabajo se presenta una experiencia práctica introduciendo IBM, utilizando como asistentes los programas Matlab[®] y/o Octave[®].

Palabras clave: Modelización matemática, IBM, Matlab[®], Octave[®].

1. Introducción

1.1 Modelos y simulación

La formación en ciencias e ingeniería queda incompleta sin una formación de calidad en matemáticas. La formación requerida a nivel institucional, y la que espera el alumno, ha de ir orientada hacia las aplicaciones, es decir hacia los modelos matemáticos (Gomez-Neves y Duarte-Teodoro, 2010). A grandes rasgos, un modelo científico es una abstracción de la realidad que nos envuelve. Se trata de representar elementos reales, utilizando un lenguaje, en general abstracto, mediante el cual el científico pretende aproximarse a la realidad para dar solución a necesidades generales de conocimiento, elaborando teorías, o quizás simplemente para resolver problemas, es decir, para satisfacer necesidades inmediatas de carácter práctico. Si el modelo científico utiliza el lenguaje matemático y herramientas matemáticas estaremos hablando de modelos matemáticos. La manera de construir el conocimiento matemático de un futuro ingeniero o científico de otras áreas, difiere de la de un estudiante que accede a un grado en matemáticas. Para el ingeniero, la cuestión de resolver problemas técnicos, incluso mediante técnicas matemáticas cuando el problema lo requiera, es mucho más importante que la del conocimiento matemático per se. El futuro ingeniero espera de las matemáticas, ya desde primer curso, incluso a nivel emocional, una “utilidad” que le permita comprender mejor los diferentes problemas que se le plantean y poder encontrar soluciones adecuadas. Quizás, en demasiadas ocasiones, el rechazo cultural hacia las matemáticas, y por lo tanto a la racionalidad del pensamiento crítico, venga motivado por la poca efectividad de los docentes a la hora de resaltar el valor instrumental de las matemáticas y por lo tanto de poner en valor el conocimiento matemático como una potente herramienta para abordar infinidad de problemas cotidianos.

La introducción de conceptos matemáticos mediante del desarrollo y estudio de modelos ha sido tratada en muchos trabajos (Khan, 2011, Ginovart, 2014) y es objeto de discusión en Jornadas y Congresos, principalmente en aquellos orientados al campo de la innovación docente. En los primeros cursos de grado parecen quedar pocas alternativas a la introducción de los modelos matemáticos desde el discurso del profesor, que describe un problema real, expone un modelo que lo representa, el estudiante experimenta con él (simula la realidad en escenarios distintos), y se demuestra al alumno la potencia de las matemáticas que han permitido resolver el modelo y obtener conclusiones útiles.

La modelización como medio para aprender matemáticas, constituye una metodología que exige tiempo y, quizás, el medio más adecuado para trabajar dicha metodología lo proporcionan las prácticas informáticas que en muchos currículums completan las asignaturas de matemáticas. Los objetivos generales serían los siguientes:

- a) Afianzar nociones teóricas desarrolladas en clase.
- b) Introducir conceptos matemáticos nuevos, no vistos en las clases teóricas.
- c) Resolver problemas de planteamiento relativamente sencillo, pero difíciles de afrontar mediante el cálculo “a mano”.
- d) Modificar la actitud del alumno ante el aprendizaje de las matemáticas, “convenciéndolo” de la utilidad de las matemáticas para entender mejor, mediante la modelación, el problema real y poder resolver hacer predicciones y/o resolver problemas.

Trabajar con modelos va íntimamente asociado a realizar pruebas con ellos, lo que viene en denominarse simulación. Según Shannon (1976), la simulación es el proceso de diseñar un modelo de un sistema real y llevar a término experiencias con él, con la finalidad de comprender el comportamiento del sistema o evaluar nuevas estrategias, dentro de los límites impuestos por un cierto criterio o un conjunto de ellos para el funcionamiento del sistema. Por otra parte Naylor (1975) define la simulación como una técnica numérica para conducir experimentos en una computadora digital. Estos experimentos comprenden ciertos tipos de relaciones matemáticas y lógicas, las cuales son necesarias para describir el comportamiento y la estructura de sistemas complejos del mundo real a través de largos periodos de tiempo.

La definición de Shannon (1976) contendría implícitamente la de Naylor (1975), dado que el primero considera simulación tanto el diseño del modelo como la evaluación de estrategias para el funcionamiento del sistema, lo cual conlleva hacer experimentos numéricos con el modelo, actividad que en muchos casos se reduce a ensayar el modelo utilizando un ordenador. Podemos encontrar otras definiciones de simulación, normalmente dependientes del contexto en que nos situemos. En este trabajo, en aras de una mayor claridad consideraremos la definición de simulación de Shannon y en el momento que se obtengan resultados numéricos, hablaremos de realizaciones. Así, por ejemplo, si tenemos una simulación, realizada mediante un modelo expresado por una ecuación en diferencias, al considerar un valor inicial o valor semilla x_0 , la sucesión que se obtiene constituye una realización de la simulación. Además, si se pretende hacer uso de los modelos para el aprendizaje de las matemáticas, lo más lógico es adoptar la definición de Shannon o cualquiera que incluya en la idea de simulación la construcción del modelo matemático y no sólo la experimentación con el mismo.

Un laboratorio de matemáticas basado en el estudio de un problema real utilizando el enfoque de los modelos IBM

Un modelo basado en agentes (agent-based model o ABM) pertenece a una clase de modelos computacionales orientados a simular las acciones e interacciones de agentes autónomos (sean entidades individuales o colectivas) con el fin de evaluar los efectos sobre el sistema en su conjunto. En ecología, a los ABM se les llama modelos basados en el individuo (individual-based model o IBM) (Grimm et al 2005). Los IBM son modelos en los que los individuos, o partes parciales que componen el sistema, son tratados de forma autónoma y, desde esta perspectiva, se obtienen los resultados globales en base a agregar los resultados individuales o parciales. Esta forma de modelar se centra en caracterizar el comportamiento del individuo, o de las partes, mediante reglas que permiten, por ejemplo, la interacción entre individuos y la de individuos con el entorno (Railsback, 2001). Los modelos IBM son adecuados, fundamentalmente, para la modelación matemática de sistemas discretos o para la modelación de sistemas continuos, previa discretización de los mismos.

1.2 Modelos matemáticos y clasificaciones usuales

Un modelo matemático es una descripción, en lenguaje matemático, de una realidad existente en un universo no necesariamente matemático.

Un modelo se caracteriza por cómo actúa frente a unos valores de entrada (inputs), modificando sus variables internas (variables de estado y otras variables auxiliares), produciendo los valores o variables de salida (outputs). Un modelo es dinámico si el tiempo es una de las entradas del sistema, es decir las variables de estado y de salida del modelo son función del tiempo. Por su parte, el modelo es estático si el tiempo no influye en el mismo.

Un modelo se dice que es discreto si sólo nos interesa conocer los valores de salida en un conjunto discreto (de cardinal finito o numerable) de instantes de tiempo. El modelo es continuo si lleva a conocer los valores de salida en todos los instantes de un intervalo temporal. Un modelo es determinista si las mismas entradas producen siempre el mismo estado y las mismas salidas, por lo que no cabe ningún tipo de aleatoriedad o dependencia del azar en el modelo. El modelo será estocástico o probabilístico si el azar interviene en el modelo, con la consiguiente posibilidad de que una misma entrada pueda llevar diversos estados y salidas, sin que éstos puedan ser predichos de antemano.

Los modelos dinámicos deterministas continuos suelen representarse mediante ecuaciones o sistemas de ecuaciones diferenciales mientras que los modelos dinámicos deterministas discretos suelen formularse a partir de ecuaciones o sistemas de ecuaciones en diferencias. Un modelo dinámico determinista, discreto o continuo, puede ser aleatorizado añadiendo a las ecuaciones que lo formulan variables de carácter aleatorio.

Los modelos deterministas, por ejemplo modelos definidos a partir de sistemas de ecuaciones en diferencias o directamente modelos definidos por expresiones analíticas, presentan un manejo teórico más sencillo que los modelos aleatorios.

2. Objetivos

Los objetivos de este trabajo se resumen en los siguientes puntos:

- Demostrar las posibilidades de Matlab y/o Octave como asistentes de cálculo para el trabajo con modelos IBM.
- Exponer una experiencia de modelización mixta, sistema determinista discreto e IBM, que permite al alumno participar más activamente en el proceso de modelización y, por lo tanto, mejorar su percepción de la utilidad de las matemáticas para resolver problemas reales.
- Valorar en qué medida el enfoque mediante modelos IBM también mejora la percepción inicial que tiene el alumno de la matemática.

3. Desarrollo de la innovación

3.1 El problema

Se considera una población de árboles de un bosque, formada por N individuos, valor que suponemos no varía en el periodo de tiempo considerado (sistema cerrado). Se detecta una plaga o enfermedad que, en principio, no produce mortalidad y que inmuniza contra la misma al árbol que la ha sufrido. Semanalmente, parte de los árboles sanos contraen la enfermedad quedando infectados, y parte de los infectados sanan volviéndose inmunes a la enfermedad.

El recorrido de aprendizaje, concepto derivado de la teoría antropológica de lo didáctico (Barquero, Bosch y Gascón, 2006; Boigues, Estruch, Roig y Vidal, 2013), que se plantea para abordar el problema se expone en la Fig 3. El problema objeto de estudio se aborda mediante un modelo discreto, que se formaliza en un sistema lineal de ecuaciones en diferencias que admite representación matricial. Esta primera aproximación permite afrontar la solución analítica y la solución numérica. Por otra parte, un modelo IBM resulta muy cercano puesto que del análisis individuo a individuo se puede llegar a la solución numérica, pero exigirá el uso de métodos de Montecarlo.

Un laboratorio de matemáticas basado en el estudio de un problema real utilizando el enfoque de los modelos IBM

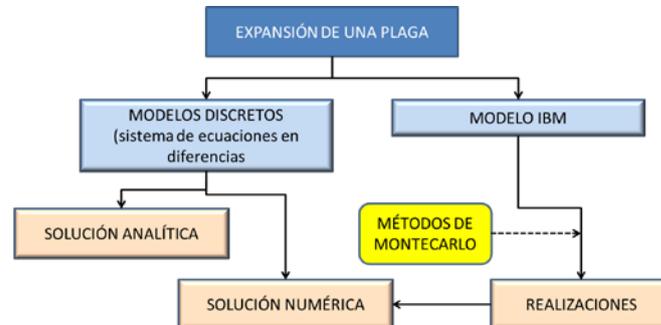


Fig. 1 mapa conceptual del recorrido de aprendizaje.

3.1.1 El modelo determinista discreto

Supondremos que se sabe que, semanalmente, un m % de la población de árboles sanos (S) contraen la enfermedad quedando infectados (I), y un c % de los infectados sanan volviéndose inmunes o resistentes a la enfermedad (R). Se supone que $S+I+R=N$. Esta hipótesis de partida puede ser útil como tema de discusión en el aula. ¿Cómo se pueden obtener los valores, obviamente aproximados, de m y c ? En la discusión aparecerán, de forma natural, conceptos estadísticos como el de muestreo, tipos de muestreo, transectos, proporciones, medias, etc. Los valores se justificarán, por lo tanto, en base a estimaciones obtenidas de estudios empíricos previos.

A continuación, se plantea al alumno una aproximación basada en un modelo matemático que se concreta en un sistema lineal de ecuaciones en diferencias que admite formulación matricial. Dicho modelo contempla la evolución de la plaga en el bosque en su conjunto. El alumno experimentará con el modelo mediante rutinas de Matlab/Octave obteniendo conclusiones. Para ello necesitará utilizar conceptos matemáticos precisos, sobre todo elementos de álgebra lineal y teoría de matrices.

Las relaciones entre variables de estado, al pasar de una etapa a otra se ilustran en la Fig. 2.

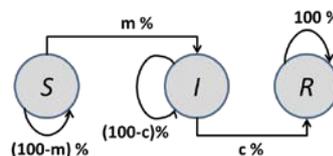


Fig. 2 Transición, entre etapas, de las variables de estado

El diagrama de flujos de la Fig 2, es un elemento fundamental en la interpretación del sistema a la hora de formular el modelo, por lo que se hará incapié en la interpretación del

mismo para formular las ecuaciones que rigen el sistema, que forman el sistema de ecuaciones en diferencias (1) para $k=0, 2, 3, \dots$,

$$\begin{aligned} S(k+1) &= \frac{100-m}{100}S(k) + 0 \cdot I(k) + 0 \cdot R(k) \\ I(k+1) &= \frac{m}{100}S(k) + \frac{100-c}{100}I(k) + 0 \cdot R(k) \\ R(k+1) &= 0 \cdot S(k) + \frac{c}{100}I(k) + 1 \cdot R(k) \end{aligned} \quad (1)$$

o, de forma equivalente, en forma matricial

$$E(k+1) = A \cdot E(k) \quad (2)$$

donde

$$E(k) = \begin{bmatrix} S(k) \\ I(k) \\ R(k) \end{bmatrix} \quad \text{y} \quad A = \begin{bmatrix} \frac{100-m}{100} & 0 & 0 \\ \frac{m}{100} & \frac{100-c}{100} & 0 \\ 0 & \frac{c}{100} & 1 \end{bmatrix}.$$

El modelo descrito es totalmente determinista y, a unos valores iniciales fijados $S(0)$, $I(0)$ y $R(0)$, corresponderá una única realización.

Utilizando Matlab/Octave, en concreto ciclos **for-end** o, alternativamente mediante potencias de matrices teniendo en cuenta que

$$E(k) = A^k \cdot E(0), \quad (3)$$

el alumno podrá simular diferentes comportamientos del sistema comprobando que, fijados m y c , la evolución de los estados sólo depende de $S(0)$, $I(0)$ y $R(0)$ y que, fijados $S(0)$, $I(0)$ y $R(0)$ los valores m y c son fundamentales para obtener distintos comportamientos del sistema simulado.

A continuación mostramos el conjuntos de códigos (m-files) que hemos empleado para simular el comportamiento del modelo determinista. Los m-files consisten en series de comandos Matlab/Octave, concatenados, que se ejecutan secuencialmente. Para crear un m-file debemos utilizar un editor de texto cualquiera, por ejemplo el que proporcionan los programas, crear un fichero de texto con las sentencias que se van a ejecutar secuencialmente y guardarlo con un nombre cualquiera (que no empiece por un número, sin espacios y sin símbolos especiales) y extensión **.m**. Los editores de Matlab y Octave añaden dicha extensión por defecto.

Un laboratorio de matemáticas basado en el estudio de un problema real utilizando el enfoque de los modelos IBM

Las ordenes que nos van a pedir las entradas son, 1-3

```
1 N= input('Número de días= ');
2 m= input('Introduce el valor de m= ');
3 c= input('Introduce el valor de c= ');
```

A continuación definimos la matriz de transición ,4, y el estado inicial, 5.

```
4 m=m/100; c=c/100; T=[1-m, 0, 0; m, 1-c, 0; 0, c, 1];
5 E0=[1752;48;0];
```

Seguidamente, escribimos las órdenes que calculan los estados desde el principio hasta la etapa N , 6-8:

```
6 A=E0;
7 for i=1:N,
8     A=[A,T^i*E0];
9 end
```

Pasamos a determinar la población de cada grupo, 10, a lo largo del tiempo, 11:

```
10 S=A(1,:); I=A(2,:); R=A(3,:);
11 t=0:N;
```

Introducimos las órdenes correspondientes a la elaboración de los dibujos, 12-18:

```
12 plot(t,S,'g'), grid
13 hold on
14 plot(t,I,'r'), plot(t,R,'b'),
15 xlabel('Tiempo (días)')
16 ylabel('Población')
17 legend('sanos','infectados','resistentes',0)
18 hold off
```

Finalmente pedimos que nos indique que a continuación se expone el estado final, 19, y evidentemente se obtiene el estado final obtenido, 20:

```
19 'Estado Final'
20 vpa(A(:,N),5)
```

Guardamos el m-file con el nombre, por ejemplo, de *plaga1.m*.

Para ejecutar el m-file, en primer lugar debemos establecer el “camino” para que el programa encuentre el fichero que crearemos, es decir tenemos que indicar al programa la carpeta donde hemos guardado el fichero. A continuación, en la ventana de comandos, escribimos el nombre del archivo que deseamos ejecutar. El proceso y resultados correspondientes a 90 días $m=1\%$ y $c=2\%$ se muestra en la Fig. 3.

```
>> plaga1
Número de días= 90
Introduce el valor de m= 1
Introduce el valor de c= 2
Introduce el estado inicial E0= [1752;48;0]
```

Estado Final
ans =
716.25
434.03
649.72

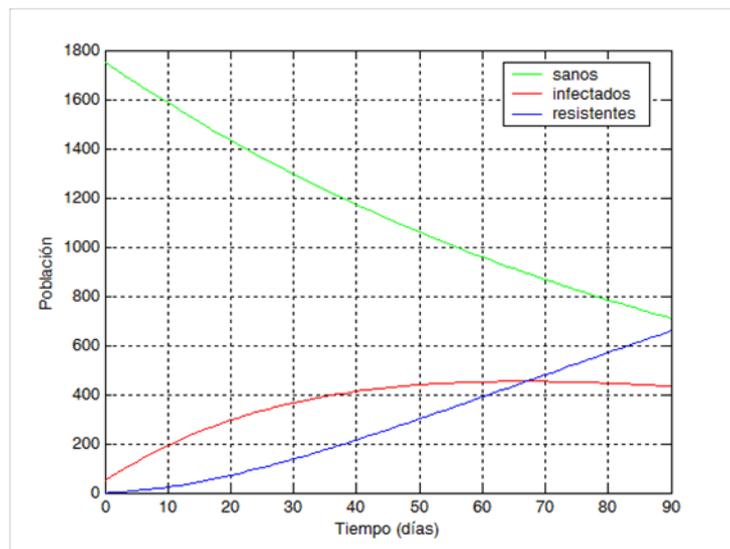


Fig.3 Resultado del m-file plaga1.m

3.1.2 El Modelo IBM

A continuación, mediante la aproximación IBM, estudiamos la evolución posible de la enfermedad en cada árbol individualmente. Se establecerá para ello, en base a una interpretación frecuencial de la probabilidad, una equivalencia entre el $m\%$ de sanos que pasan a infectados, que sería equivalente a que al pasar de una semana a otra, la probabilidad de que un árbol sano se infecte es de $m/100$. Por otra parte, que un $c\%$ de árboles infectados pasen a ser resistentes la siguiente semana sería equivalente a que la probabilidad de que un árbol infectado pase a ser resistente es de $c/100$. De esta forma se

Un laboratorio de matemáticas basado en el estudio de un problema real utilizando el enfoque de los modelos IBM

puede explicar la evolución de una semana a otra para un árbol cualquiera, lo cual se resume en la Fig. 4, donde los valores en cada flecha indican probabilidades.

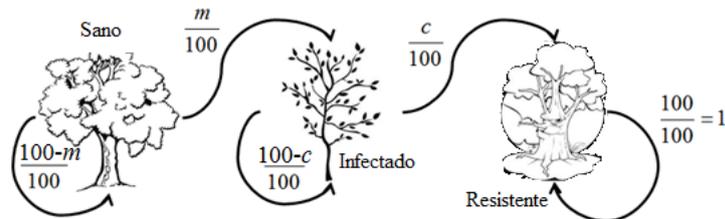


Fig.4 Transición, entre etapas, de la situación del estado de un árbol cualquiera.

Obviamente esta descripción, árbol a árbol, lleva a un modelo estocástico, donde al pasar de una etapa a otra, no siempre, exactamente un $m\%$ de los árboles sanos van a infectarse y que un $c\%$ de los árboles infectados pasarán a ser inmunes. En la realidad se pueden dar muchísimas situaciones a la hora de describir la evolución del bosque completo.

Utilizando un método de Montecarlo para las simulaciones, árbol a árbol, llegaremos a explicar el comportamiento de la plaga en todo el bosque mediante la agregación de los resultados individuo a individuo.

3.1.2.1 Procedimiento para la simulación IBM con Matlab[®]/Octave[®]

Supondremos que los árboles están ordenados en un rectángulo con F filas y K columnas, siendo el total de la población $N=F \times K$. Se definirá una matriz $a = (a_{ij})$, $i=1,2,\dots,F$, $j=1,\dots,K$. Para describir el grupo al que pertenece el individuo que ocupa una posición arbitraria (i,j) , supondremos que $a(i,j)=1$ si el árbol es resistente, $a(i,j)=2$ si es sano y $a(i,j)=3$ si está infectado.

Las entradas del modelo son: F (filas de la matriz), K (columnas de la matriz), S , I , R (número inicial de árboles susceptibles, infectados y resistentes, respectivamente), m (tasa de infección), c (tasa de resistencia) y número de etapas, *Etapas*.

Si en una etapa se tiene una matriz \mathbf{a} , en la siguiente pueden cambiar los elementos que no sean resistentes. El valor $\mathbf{a}(\mathbf{i},\mathbf{j})=1$ permanecerá fijo, mientras que si en una etapa $\mathbf{a}(\mathbf{i},\mathbf{j})=2$ (susceptible), en la siguiente puede permanecer igual o cambiar a $\mathbf{a}(\mathbf{i},\mathbf{j})=3$ (infectado) y si en una etapa $\mathbf{a}(\mathbf{i},\mathbf{j})=3$ (infectado), en la siguiente o permanece igual o bien $\mathbf{a}(\mathbf{i},\mathbf{j})=1$ (resistente). Pero, en los dos últimos casos, se establecerá si se produce o no el cambio mediante simulación probabilística, es decir generando valores aleatorios. Para ello se puede utilizar el comando **rand** de Matlab/Octave, que genera un valor real pseudo-aleatorio distribuido uniforme en el intervalo de extremos 0 y 1. Pero hay que tener en cuenta que Matlab restablece la secuencia de valores pseudo-aleatorios a un valor inicial fijo en el arranque, y la secuencia de números vuelve a ser la misma, a menos que se

cambie dicho estado inicial. Para aleatorizar más el proceso se puede utilizar antes de generar valores la sentencia de Matlab/Octave `rand('state',sum(100*clock))`, que establece el valor inicial dependiendo de la hora. La orden de Matlab/Octave `rand(n,1)` genera n valores pseudo-aleatorios y `rand(F,K)` genera una matriz $F \times K$ de valores pseudo-aleatorios, siempre distribuidos uniforme en el intervalo de extremos 0 y 1.

Si en una etapa $a(i,j)=2$ (individuo en (i,j) sano), se genera un valor aleatorio, u , y si $u < m/100$, se supone que el árbol se infecta y por tanto en la siguiente etapa $a(i,j)=3$; en otro caso $a(i,j)=2$. De forma análoga, si $a(i,j)=3$ (infectado) se genera un valor aleatorio u , y si $u < c/100$, suponemos que el árbol se vuelve resistente, $a(i,j)=1$; en otro caso $a(i,j)=3$. Para mayor comodidad, se puede generar una matriz $F \times K$ de valores aleatorios, `aleat`, y hacer uso de los valores aleatorios que ocupen la ubicación de un individuo sano o infectado, `aleat(i,j)`.

En cada etapa se actualiza el número total de sanos, infectados y resistentes, contando cuántos elementos con valor 1, 2 y 3 hay en la nueva matriz.

La aproximación IBM lleva, de forma natural, al origen del modelo matricial descrito en un principio. La simulación IBM es realizada en dos fases: Primero, a mano, utilizando una ficha de trabajo para el caso sencillo de un bosque con 9 árboles representado por una matrix 3×3 (Fig. 5).



Fig.5 Simulación de una transición a partir de una distribución fijada en la etapa 0

Posteriormente, utilizando Matlab/Octave y una rutina sencilla desarrollada ex profeso, se realizan simulaciones en un bosque más realista, con muchos más árboles. La agregación de la información obtenida individualmente para cada árbol permitirá estimar la expansión global de la plaga en el bosque. En primer lugar creamos tres funciones:

a) `simule.m`, que representa los tipos de arboles.

```
function simule (A)
[F,K]=size(A);
```

Un laboratorio de matemáticas basado en el estudio de un problema real utilizando el enfoque de los modelos IBM

```
for i=1:F
    for j=1:K
        if A(i,j)==1, plot(i,j,'bo')
        elseif A(i,j)==2, plot(i,j,'go')
        else plot(i, j, 'ro')
        end
    end
end
hold on
end
end
```

b) *cuenta.m*, que cuenta los arboles que hay de cada clase.

```
function Res=cuenta(A)
[F,K]=size(A); S=0; l=0; R=0;
for i=1:F
    for j=1:K
        if A(i,j)==1, R=R+1;
        elseif A(i,j)==2, S=S+1;
        else l=l+1;
        end
    end
end
end
Res=[S;l;R];
```

c) *novaetapa.m*, que simula el paso de una etapa a una nueva.

```
function A= novaetapa(A,m,c)
m=m/100; c=c/100;
[F,K]= size(A);
aleat=rand(F,K);
for i=1:F
    for j=1:K
        if A(i,j)==1 %Resistente se queda igual
            A(i,j)=A(i,j);
        elseif and (A(i,j)==2, aleat(i,j)<m) %sano que se infecta
            A(i,j)=3;
        elseif and (A(i,j)==2, aleat(i,j)>=m) %sano que no cambia
            A(i,j)=A(i,j);
        elseif and (A(i,j)==3, aleat(i,j)<c) %infectado que cambia a resistente
            A(i,j)=1;
        elseif and (A(i,j)==3, aleat(i,j)>=c) %infectado que se queda igual
            A(i,j)=A(i,j);
        end
    end
end
end
```

Segídamente se describe el M-file creado para simular el IBM, denominado *plagas2.m*.

Introduzcamos las ordenes que nos van a pedir las entradas, 1-4

```
1 A0= input('Matriz posición de la plaga= ');
2 m= input('Introduce la tasa de infección m= ');
3 c= input('Introduce la tasa de resistencia c= ');
4 etapas= input('Introduce el número de etapas= ');
```

a continuación se obtienen los resultados del proceso, 5-10, y las representaciones gráficas, 11-17.

```
5 P=A0; P0=cuenta(A0);
6 for k=1:etapas
7     P= novaetapa(P,m,c);
8     Pob=cuenta(P); P0=[P0, Pob];
9 end
10 t=0:1:etapas; S=P0(1,:); I= P0(2,:); R=P0(3, :);
11 subplot(2,2,1), simule(A0),grid on, title('Distribución inicial')
12 subplot(2,2,2), plot(t,S,'g'), grid on, hold on, plot(t,I,'r'), plot(t,R,'b')
13 legend('sanos','infectados','resistentes')
14 xlabel('días'), ylabel('árboles'), title('Evolución de la población')
15 subplot(2,2,3), simule(P)
16 title('Distribución final')
17 subplot(2,2,4), pcolor(P),shading interp, title('Distribución final')
```

Finalmente se tienen las órdenes que proporcionan como salida el estado final, 18-21

```
18 clc, disp('ESTADO FINAL IBM')
19 disp('Sanos= '), disp(P0(1,etapas+1))
20 disp('Infectados= '),disp(P0(2,etapas+1))
21 disp('Resistentes= '),disp(P0(3,etapas+1))
```

Simulamos la evolución de la plaga a lo largo de 90 días, partiendo de una matriz inicial, **A0** (Fig. 6), de 60x20:

```
>> plagas2
```

```
Matriz posición de la plaga= A0
```

```
Introduce la tasa de infección m= 1
```

Un laboratorio de matemáticas basado en el estudio de un problema real utilizando el enfoque de los modelos IBM

Introduce la tasa de resistencia $c = 2$

Introduce el número de etapas = 90

El resultado es un posible número de sanos infectados y resistentes en la etapa final (etapa 90 en el ejemplo) junto a 4 gráficos que, de izquierda a derecha y de arriba abajo muestran: la distribución inicial y estado de los árboles, la evolución de las poblaciones de sanos infectados y resistentes desde la etapa inicial a la final (90 en este ejemplo), la distribución final y un mapa de colores que da una idea de la distribución de las poblaciones, difuminada por zonas, en la etapa final. Un ejemplo se muestra en la Fig. 6.:

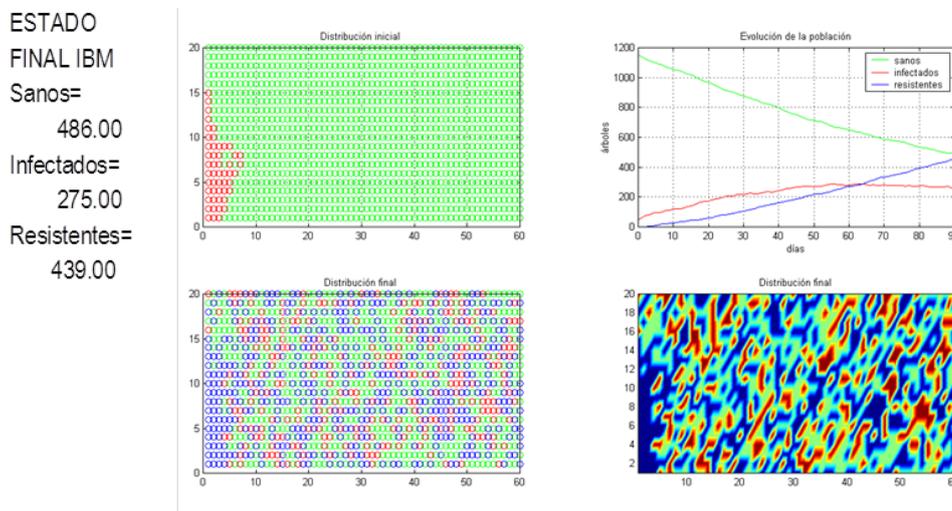


Fig.6 Resultado del m-file *plaga2.m*

4. Resultados

La experiencia se ha desarrollado en dos grupos de prácticas de Matemáticas del Grado en Ciencias Ambientales y 5 grupos del grado de Ingeniería Química de la Escuela Técnica Superior de Ingenieros Industriales, de la Universitat Politècnica de València. Cerca de 200 estudiantes de primer curso participaron en la experiencia, que se desarrolló en tres sesiones de hora y media en un laboratorio informático. En la primera sesión se trabajó el modelo determinista y en la segunda el modelo IBM. La tercera sesión dedicó al desarrollo de ejemplos que implicaban modificaciones del modelo inicial.

Al principio de la experiencia se pasó un test para medir las creencias y actitudes frente a las matemáticas en general. Al finalizar la experiencia se pasó otro test similar al primero. Si bien las valoraciones que hacen los alumnos de la experiencia son diversas y desiguales, se observa que un número importante de ellos, alrededor del 27%, se manifiestan su

motivación ante el desafío de resolver el problema planteado, despertando en los mismos el interés por aplicar la técnica de modelización IBM a otros problemas o contextos.

5. Conclusiones

El uso de la aproximación IBM en problemas de modelización posibilita un aprendizaje más autónomo y completo del alumno a la hora de adquirir la competencia de construir modelos que representen situaciones reales. Por otra parte, los modelos IBM sirven de trampolín para afrontar otros tipos de modelización más complejos, los cuales se entienden mejor si, en paralelo a los planteamientos clásicos, se trabajan desde la perspectiva de los IBM.

6. Referencias

BARQUERO, B.; BOSCH, M. Y GASCÓN, J. (2006). Una nova Organització del curriculum de matemàtiques de primer curs universitari de ciències: els recorreguts d'Estudi i Investigació. IV Congreso Internacional Docencia Universitaria e Innovación.

BOIGUES, F.J.; ESTRUCH V.D.; ROIG B. Y VIDAL, A. (2013). Una propuesta de recorrido de estudio e investigación(REI): Diseño, simulación y decisión de una estrategia de pesca sostenible. *Modelling in Science Education and Learning* 6(2), 5-19.

GINOVAR, M. (2014). Discovering the power of individual-Based Modelling in teaching and learning: The study of a predator-prey system. *Journal of Science Education and Technology* 23(4), 496-513.

GOMEZ-NEVES, R. Y DUARTE-TEODORO, V. (2010). Enhancing science and mathematics education with computacional modelling. *Journal Mathe Model appl 1: 2-15*.

GRIMM V. Y RAILSBACK, S.F. (2005). *Individual-based Modeling and Ecology*. Princeton University Press. 485 pp.

KHAN, S. (2011). New pedagogies on teaching science with computer simulations. *Journal Science education Technology* 20, 215-232.

NAYLOR, T. H. (1975). *Técnicas de simulación en computadoras*, Ed. Limusa

SHANNON, R. Y JOHANNES, J. D. (1976). Systems simulation: the art and science. *IEEE Transactions on Systems, Man and Cybernetics* 6(10). pp. 723-724.

RAILSBACK S.F. (2001). Concepts from complex adaptive systems as a framework for individual-based modelling. *Ecological Modelling* 139, 47-62.