DE Oddity

Daniel Fernández Martínez

Enero 2020

Resumen

En esta memoria se recoge el desarrollo de la parte del "examen" final destinada a la resolución de un PVI con un método numérico que satisfaga la condición de error impuesta en el enunciado. En el primer caso, la Ecuación Diferencial es lineal y, por tanto, tiene solución exacta. En los dos últimos casos, la Ecuación pasa a ser no lineal y no podemos comparar la solución con su solución exacta, pues esta no existe.

Índice

1. Enunciado

Tarea Navidad 2020. A DE oddity

Este ejercicio es común. Se considera la ED $x' = 10(x - t^2 + 0.2t - 1)$ con la CI $x(0) = 1$. Dado que se trata de una ED lineal podemos resolverla analíticamente. La solución general es

$$x = t^2 + 1 + ce^{10t} \tag{1}$$

Luego la solución exacta del PVI es la parábola $x_{\text{ex}}(t) = t^2 + 1$. Si uno prueba un método numérico para aproximar la solución, incluso con pasos bastante pequeños, el error se vuelve enseguida muy grande. La razón es clara: en cuanto cometemos un error en la aproximación pasamos de $c = 0$ a $c \neq 0$ en la solución general (1), nos "salimos" de la parabola y el término exponencial "agobia" a la solución exacta.

El ejercicio consiste en proponer y analizar un método numérico que "garantice", en la medida de lo posible, que la solución obtenida es fiable hasta $t = 1.5$ con un error de digamos 0.5 (o sea, que $|x_{\text{num}}(1.5) - x_{\text{ex}}(1.5)| \leq 0.5$.

Obsérvese que en este caso podemos saber lo que ocurre muy bien porque tenemos una solución exacta para comparar. Ahora viene lo malo y es que pongamos nuestro método a trabajar para resolver los problemas:

- $x' = 10(x^a - t^2 + 0.2t - 1)$ con la CI $x(0) = 1$.

Primero hacer $a = 0.9$ y luego $a = 1.1$. Se entiende que debemos seguir garantizando la misma cota de error que antes, pero ahora no sabemos la solución exacta, ni siquiera qué forma tiene.

2. Resolución

2.1. $a = 1$

La ecuación a resolver es, por tanto: $x' = 10(x - t^2 + 0{,}2t - 1)$, $x(0) = 1$

Dicha ecuación es lineal, y tiene como solución exacta:

$$x_{ex}(t) = t^2 + 1$$

Deberemos de proponer un método numérico que garantice que la solución satisface la condición de error hasta $t = 1{,}5$.

Para ello, se ha elegido como método numérico un método de un sólo paso: el Runge - Kutta de cuarto orden, más conocido como Runge - Kutta a secas.

Este utiliza cuatro valores auxiliares en cada paso, en los que se ha de calcular la función f cuatro veces.

$$p_1 = \tau f(u_n, t_n)$$
$$p_2 = \tau f(u_n + \frac{p_1}{2}, t_{n+\frac{1}{2}})$$
$$p_3 = \tau f(u_n + \frac{p_2}{2}, t_{n+\frac{1}{2}})$$
$$p_4 = \tau f(u_n + p_3, t_{n+1})$$

u_{n+1} se obtiene a partir de los cuatro valores anteriores:

$$u_{n+1} = u_n + \frac{1}{6}(p_1 + 2p_2 + 2p_3 + p_4)$$

La implementación en MATLAB es casi calcada a la formulación anterior.

```
% Metodo de Runge - Kutta de 4   Orden
for i=1:(length(x)-1)

    p1 = F_tx(t(i),x(i));
    p2 = F_tx(t(i)+0.5*h,x(i)+0.5*h*p1);
    p3 = F_tx((t(i)+0.5*h),(x(i)+0.5*h*p2));
    p4 = F_tx((t(i)+h),(x(i)+p3*h));

    x(i+1) = x(i) + (1/6)*(p1+2*p2+2*p3+p4)*h;
end
```

Introduciendo los datos de partida del problema:

```
% Parametros auxiliares:
h = 0.01; % Paso temporal
t_ini = 0; % Extremo izquierdo del intervalo
t_fin = 1.5; % Extremo derecho del intervalo
t=t_ini:h:t_fin; % Intervalo
t_0=1; % Condicion inicial (t)
x_0=1; % Condicion inicial (x)
```

```
x=zeros(1,length(t)); % Vector x
x(t_0)=x_0; % Condicion inicial

% ODE:
a = 1; % a
F_tx=@(t,x) 10*(x^a - t^2 + 0.2*t - 1);

% Soluci n exacta
t_exact = linspace(0,1.5);
x_exact = t_exact.^2 + 1;
```

Así, la implementación en MATLAB completa quedaría:

```
%% DE oddity - Daniel Fern ndez Mart nez
% Se considera la ED x' = 10(x^a + t^2 + 0.2t - 1) con la CI ...
    x(0) = 1.
% Proponer y analizar un m todo num rico que    garantice   , ...
    en la medida
% de lo posible, que la soluci n obtenida es fiable hasta t = ...
    1.5 con un
% error de digamos 0.5 (o sea que |xnum(1.5) ? xex(1.5)| ? 0.5

% Par metros auxiliares:
h = 0.01; % Paso temporal
t_ini = 0; % Extremo izquierdo del intervalo
t_fin = 1.5; % Extremo derecho del intervalo
t=t_ini:h:t_fin; % Intervalo
t_0=1; % Condici n inicial (t)
x_0=1; % Condici n inicial (x)
x=zeros(1,length(t)); % Vector x
x(t_0)=x_0; % Condici n inicial

% ODE:
a = 1; % a
F_tx=@(t,x) 10*(x^a - t^2 + 0.2*t - 1);

% Soluci n exacta
t_exact = linspace(0,1.5);
x_exact = t_exact.^2 + 1;

% M todo de Runge - Kutta de 4   Orden
for i=1:(length(x)-1)

    p1 = F_tx(t(i),x(i));
    p2 = F_tx(t(i)+0.5*h,x(i)+0.5*h*p1);
    p3 = F_tx((t(i)+0.5*h),(x(i)+0.5*h*p2));
    p4 = F_tx((t(i)+h),(x(i)+p3*h));

    x(i+1) = x(i) + (1/6)*(p1+2*p2+2*p3+p4)*h;
end

% C lculo del error cometido
disp('Error cometido:')
err = abs(x(length(x)) - x_exact(length(t_exact)))

% Visualizaci n
```

```
figure;
plot(t,real(x));
hold on;
plot(t_exact, x_exact)
hold off;
title(['Soluci n num rica (a = ', num2str(a), ', h = ', ...
    num2str(h), ') | Err(1.5) = ', num2str(err)]);
legend('Runge-Kutta', 'Exacta');
```

Con dicha implementación, se obtienen los siguientes resultados, para distintos pasos temporales, h:

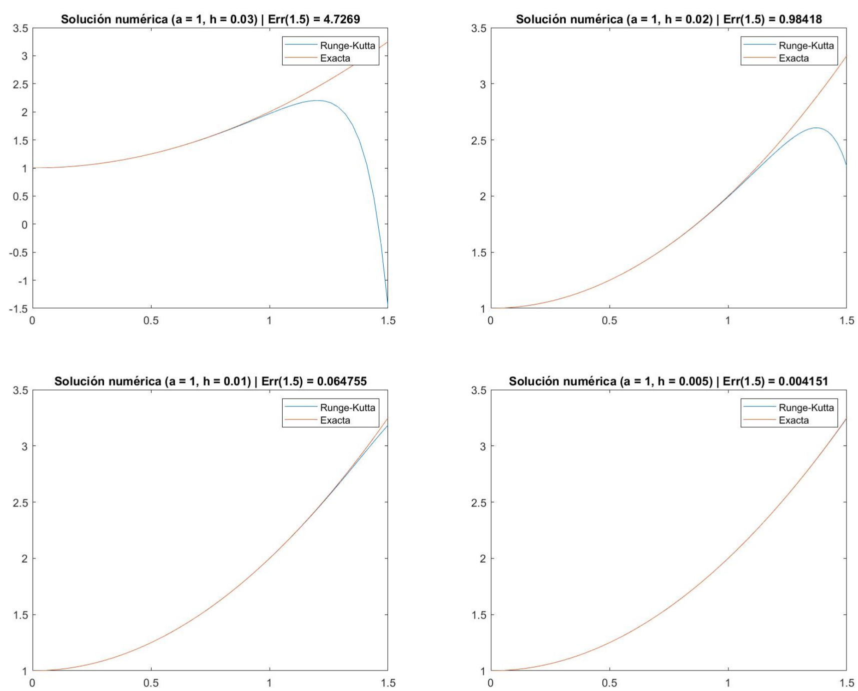

Figura 1: Gráficas de la solución numérica y la solución exacta para distintos pasos temporales

Se puede comprobar que a partir de $h = 0{,}01$ se consigue sastisfacer la condición del error. Si seguimos disminuyendo el paso temporal, la solución se ajustará cada vez más a la solución exacta.

En los siguientes subapartados se tomarán valores de a que harán que la ecuación sea no lineal, de la cual no tenemos solución exacta con la que comparar.

2.2. $a = 0{,}9$

La ecuación a resolver es ahora, por tanto: $x' = 10(x - t^2 + 0{,}2t - 1)$, $x(0) = 1$. En consecuencia, el enfoque ahora tendrá que ser un poco distinto.

Trataremos de ir dismiuyendo el paso temporal para determinar si la solución converge hacia una función concreta.

El código empleado en este caso es el que se muestra a continuación.

```
%% DE oddity - Daniel Fern ndez Mart nez
% Se considera la ED x' = 10(x^a + t^2 + 0.2t - 1) con la CI ...
    x(0) = 1.
% Proponer y analizar un m todo num rico que   garantice  , ...
    en la medida
% de lo posible, que la soluci n obtenida es fiable hasta t = ...
    1.5 con un
% error de digamos 0.5 (o sea que |xnum(1.5) ? xex(1.5)| ? 0.5

% Caso 2: a = 0.9

% Par metros auxiliares:
h = 0.001; % Paso temporal
t_ini = 0; % Extremo izquierdo del intervalo
t_fin = 1.5; % Extremo derecho del intervalo
t=t_ini:h:t_fin; % Intervalo
t_0=1; % Condici n inicial (t)
x_0=1; % Condici n inicial (x)
x=zeros(1,length(t)); % Vector x
x(t_0)=x_0; % Condici n inicial

% ODE:
a = 0.9; % a
F_tx=@(t,x) 10*(x^a - t^2 + 0.2*t - 1);

% M todo de Runge - Kutta de 4  Orden
for i=1:(length(x)-1)

    p1 = F_tx(t(i),x(i));
    p2 = F_tx(t(i)+0.5*h,x(i)+0.5*h*p1);
    p3 = F_tx((t(i)+0.5*h),(x(i)+0.5*h*p2));
    p4 = F_tx((t(i)+h),(x(i)+p3*h));

    x(i+1) = x(i) + (1/6)*(p1+2*p2+2*p3+p4)*h;
end

% Visualizaci n
figure;
plot(t,real(x));
hold on;
plot(t, imag(x));
hold off;
title(['Soluci n num rica (a = ', num2str(a), ', h = ', ...
    num2str(h),')']);
legend('Real','Imaginaria');
```

Cabe destacar que para esta a, se obtienen soluciones reales e imaginarias.

Con dicha implementación, se obtienen los siguientes resultados, para distintos pasos temporales, h:

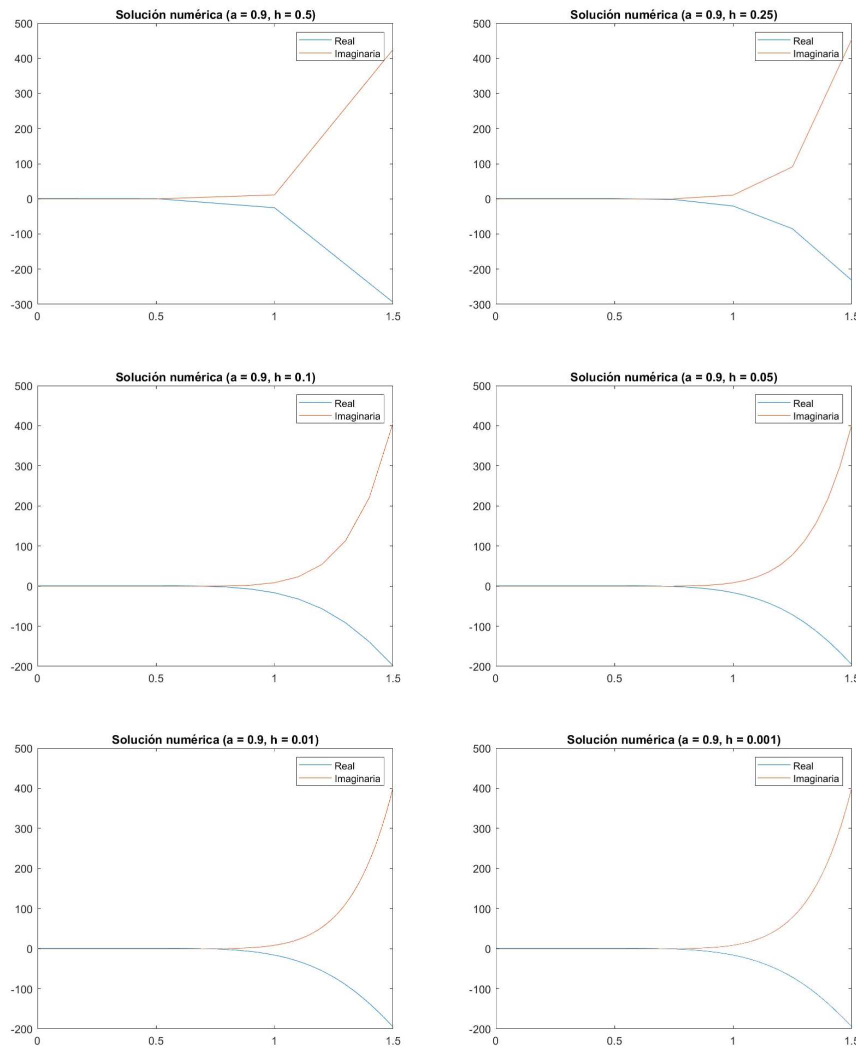

Figura 2: Gráficas de las soluciones numéricas para distintos pasos temporales

Se observa que todas las soluciones, a partir de $h = 0{,}1$, convergen hacia una misma función, que es la solución de la ecuación diferencial. Por tanto, si seguimos disminuyendo el paso, la solución no se va a modificar, por lo que estaríamos cumpliendo con la condición de error.

2.3. $a = 1{,}1$

De igual forma, se pueden obtener las soluciones (esta vez únicamente reales) para $a = 1{,}1$: La ecuación a resolver es, por tanto: $x' = 10(x^1{,}1 - t^2 + 0{,}2t - 1)$, $x(0) = 1$. El código para este caso, quedaría:

```
%% DE oddity - Daniel Fern ndez Mart nez
% Se considera la ED x' = 10(x^a + t^2 + 0.2t - 1) con la CI ...
    x(0) = 1.
% Proponer y analizar un m todo num rico que    garantice   , ...
    en la medida
% de lo posible, que la soluci n obtenida es fiable hasta t = ...
    1.5 con un
% error de digamos 0.5 (o sea que |xnum(1.5) ? xex(1.5)| ? 0.5

% Caso 3: a = 1.1

% Par metros auxiliares:
h = 0.2; % Paso temporal
t_ini = 0; % Extremo izquierdo del intervalo
t_fin = 1.5; % Extremo derecho del intervalo
t=t_ini:h:t_fin; % Intervalo
t_0=1; % Condici n inicial (t)
x_0=1; % Condici n inicial (x)
x=zeros(1,length(t)); % Vector x
x(t_0)=x_0; % Condici n inicial

% ODE:
a = 1.1; % a
F_tx=@(t,x) 10*(x^a - t^2 + 0.2*t - 1);

% M todo de Runge - Kutta de 4   Orden
for i=1:(length(x)-1)

    p1 = F_tx(t(i),x(i));
    p2 = F_tx(t(i)+0.5*h,x(i)+0.5*h*p1);
    p3 = F_tx((t(i)+0.5*h),(x(i)+0.5*h*p2));
    p4 = F_tx((t(i)+h),(x(i)+p3*h));

    x(i+1) = x(i) + (1/6)*(p1+2*p2+2*p3+p4)*h;
end

% Visualizaci n
figure;
plot(t,x);
title(['Soluci n num rica (a = ', num2str(a), ', h = ', ...
    num2str(h),')']);
```

En este caso, vemos en la Figura 3 que tenemos que disminuir el paso temporal hasta la centésima, $h = 0{,}01$, para comenzar a ver un comportamiento idéntico en las sucesivas soluciones. Por tanto, al igual que en el caso anterior, la solución converge hacia una función y, por tanto, se puede deducir que estaríamos satisfaciendo el requisito del error.

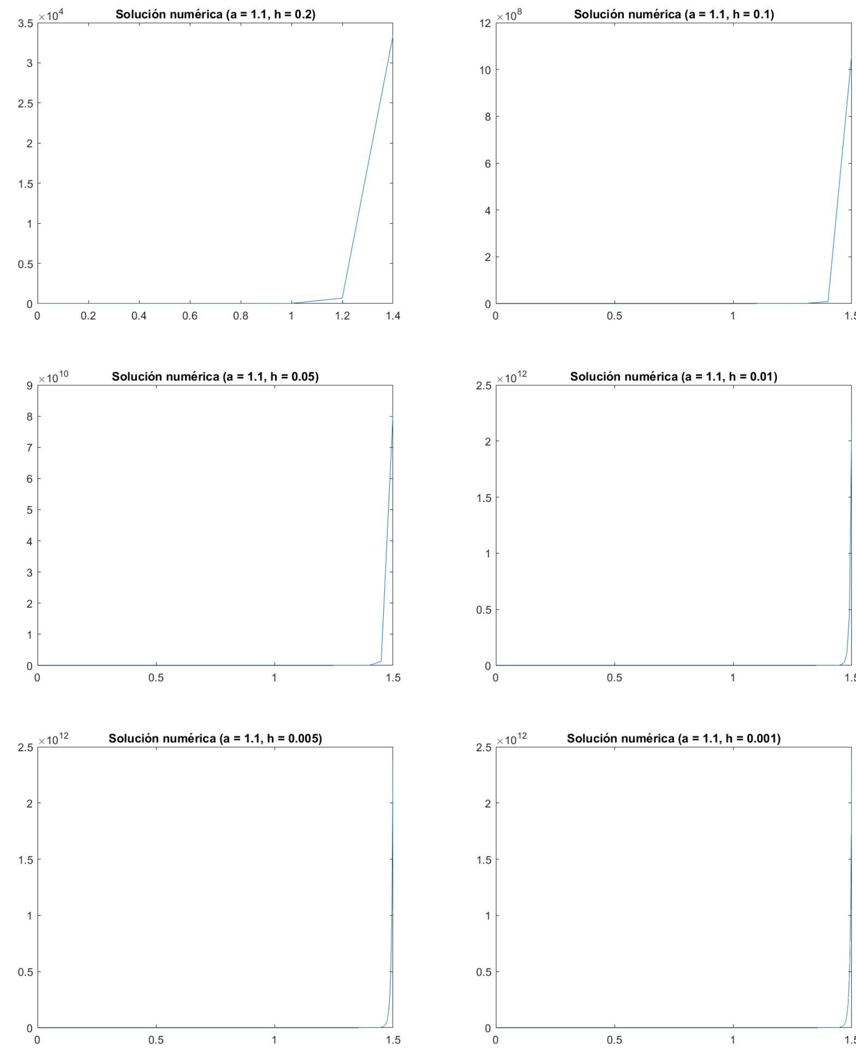

Figura 3: Gráficas de las soluciones numéricas para distintos pasos temporales