

Trabajo Práctico Nro. 2

Métodos Numéricos - 2019

Grupo 1:

Mestanza Joaquín

Müller Malena

Nowik Ariel

Regueira Marcelo

Reller Thomas

June 6, 2019

1 RESUMEN

Es interesante plantearse cuándo se requiere la utilización de métodos numéricos para la resolución de problemas, en este caso ecuaciones diferenciales ordinarias (en el tratado del presente trabajo). No sólo se las utiliza cuando las soluciones analíticas no pueden ser expresadas por las clásicas funciones fundamentales; sino que a su vez se tiene como ejemplo el paper del que se tomaron las EDOs que se estudian en este trabajo práctico, en donde se expone otra razón para la utilización de análisis numérico: a boca del propio autor, comenta que en el estado estacionario la solución analítica para encontrar la raíz consiste en más de 200 términos con los que realizar álgebra, haciéndolo engorroso en la práctica. Además pueden generar errores numéricos ligados a las operaciones algebraicas en representación de tipo float. Por ende, en este caso, se optó por encontrar las raíces de forma numérica, conociendo previamente que el sistema tiene un único punto fijo en la región local del espacio parametrizado explorado (Model Analysis and Simulation Results en el paper).

Tras reconocer que Heun está contenido en el segundo orden de Runge-Kutta, resulta entonces como pregunta natural cuestionarse cuándo es conveniente parar de escalar en el orden, es decir, cuándo la eficiencia empieza a bajar de manera mucho más marcada en contraposición a la precisión lograda. La respuesta en realidad depende de la naturaleza del problema. Pero como regla general (obviamente existen excepciones), es conveniente usar un orden mayor cuando se está muy limitado a errores de redondeo, porque mayor orden requiere menos pasos. Es decir, menos evaluaciones de funciones, cuentas, y demás). En este trabajo se usó Heun por consigna.

Se comenta brevemente la manera de aproximar de Heun, que se ve reflejada los códigos más adelante. La misma es un modelo superador de Euler, de mayor precisión. Euler lo que hace es tomar la línea tangente a la función al principio del intervalo asumiendo que el paso es suficientemente chico como para aproximar a la tangente. Sin embargo, aún cuando el paso sea verdaderamente muy chico, el error se va acumulando en muchos casos considerablemente. Heun considera las líneas tangentes a la curva solución en ambos puntos que limitan el intervalo, uno que subestima y otro que sobreestima la coordenada vertical ideal, y sobre ellos realiza una ponderación equitativa. En el propio método se nota que la piedra fundacional del algoritmo está emprendida por Euler como pionero.

Basándonos en el desarrollo numérico del artículo de *Lemaire et al. [1]: "Modeling the interactions between osteoblast and osteoclast activities in bone remodeling"*, en este trabajo se resuelven las siguientes ecuaciones diferenciales ordinarias implementando el método de Heun.

$$\begin{aligned}\frac{dR}{dt} &= D_R \cdot \pi_C - \frac{D_B}{\pi_C} \cdot R + I_R \\ \frac{dB}{dt} &= \frac{D_B}{\pi_C} \cdot R - k_B \cdot B + I_B \\ \frac{dC}{dt} &= D_C \cdot \pi_L - D_A \cdot \pi_C \cdot C + I_C\end{aligned}$$

Las constantes de estas ecuaciones se encuentran especificadas en el Paper mencionado. A partir de estas tres ecuaciones diferenciales ordinarias, se graficaron nueve figuras que cumplen con los parámetros especificados en el enunciado de la consigna del presente trabajo práctico.

2 IMPLEMENTACIÓN

Sobre cuestiones ligadas a la implementación: se usó un método iterativo para encontrar el máximo error absoluto admisible. El mismo no es de los más eficientes que se pudieron haber implementado ya que,

por ejemplo, podría haberse aumentado el orden de eficiencia realizando una búsqueda binaria para localizar un intervalo chico. Sin embargo, no se lo consideró algo que amerite demasiado trabajo de optimización, porque no logra una mejora sustancial y de esta forma el código resulta más legible, por lo que se eligió este último criterio. A continuación se presentan los códigos implementados para obtener los gráficos que se encuentran más abajo.

```

1 function [x t]=heun(f,x0,t0,tf,n,par)
2     h=(tf-t0)/n;
3     t=linspace(t0,tf,n);
4     x(1,:)=x0;
5     for i=1:n-1
6         k1(i,:)=f(t(i),x(i,:),par);
7         k2(i,:)=f(t(i)+h,x(i,:)+k1(i,:),par);
8         x(i+1,:)=x(i,:)+(h/2)*(k1(i,:)+k2(i,:));
9     end

```

Listing 1: heun.m

```

1 function [x t]=miode(f,x0,t0,tf,dtmax,tol,par)
2     n = ceil((tf-t0)/dtmax);
3     err = 1;
4
5     while (err > tol)
6         [x1 t1] = heun(f,x0,t0,tf,n,par);
7         [x2 t2] = heun(f,x0,t0,tf,n*2,par);
8         err = max(abs(x1(tf)-x2(tf)));
9         n = n + 1;
10    end
11
12    [x t] = heun(f,x0,t0,tf,n-1,par);

```

Listing 2: miodo.m

```

1 f=@(t,y,p) 2*t;
2
3 [x t] = miodo(f, 1, -2, 2, 1, 10^-1,'a');
4
5 plot(t,x)

```

Listing 3: ejemplo.m

```

1 clear all;
2 clc;
3
4 I_R=@(t,p) strcmp(p,'a3')*(0.0001).*(20<=t && t<=80)+strcmp(p,'b3')*(-0.00012).*(20<=t && t<=80);
5
6 I_B=@(t,p) strcmp(p,'a1')*(0.0001).*(20<=t && t<=80)+strcmp(p,'b1')*(-8.3*10^-5).*(20<=t && t<=80);
7
8 I_C=@(t,p) strcmp(p,'a2')*(0.0001).*(20<=t && t<=80)+strcmp(p,'b2')*(-0.00029).*(20<=t && t<=80);
9
10 I_P=@(t,p) strcmp(p,'c1')*(1000).*(20<=t && t<=80);
11
12 I_0=@(t,p) strcmp(p,'c2')*(2.5*10^5).*(20<=t && t<=80);
13
14 I_L=@(t,p) strcmp(p,'c3')*(10^4).*(t>=20);
15

```

```

16 % constantes
17 k1 = 10^(-2);
18 k2 = 10;
19 f0 = 0.05;
20 d_B = 0.7;
21 Cs = 5*10^(-3);
22 k3 = 5.8 * 10^(-4);
23 k4 = 1.7 * 10^(-2);
24 k0 = 0.35;
25 K = 10;
26 klp = 3*10^6;
27 kop = 2*10^5;
28 r_L = 10^3;
29 k_P = 86;
30 S_P = 250;
31 k6 = 3;
32 k5 = 0.02;
33
34 P0 = S_P / k_P;
35 Ps = k6 / k5;
36
37 D_R = 7*10^(-4);
38 D_B = f0 * d_B;
39 k_B = 0.189;
40 D_C = 2.1*10^(-3);
41 D_A = 0.7;
42 R = 0.0007734;
43 B = 0.0007282;
44 C = 0.0009127;
45
46 pi_C = @(t,y,p) (y(3) + f0 * Cs)/(y(3) + Cs);
47 P_ = @(t,p) I_P(t,p) / k_P;
48 pi_p = @(t,p) (P_(t,p) + P0)/(P_(t,p) + Ps);
49 pi_L = @(t,y,p) (k3/k4)*((klp * pi_p(t,p) * y(2))/(1 + (k3*K/k4) + (k1/(k2*k0))*((kop/pi_p(t,p))*y
    (1)) + I_0(t,p) ))) * (1 + (I_L(t,p)/r_L));
50
51
52 f=@(t,y,p) [
53     D_R*pi_C(t,y,p)-(D_B/pi_C(t,y,p))*y(1)+I_R(t,p),D_B/pi_C(t,y,p)*y(1)-k_B*y(2)+I_B(t,p),D_C*pi_L(t
    ,y,p)-D_A*pi_C(t,y,p)*y(3)+I_C(t,p)
54 ];
55
56
57 [x t] = miode(f, [R B C], 0, 140, 1, 10^-6,'a1');
58 plot(t, x)
59 [x t] = miode(f, [R B C], 0, 140, 1, 10^-6,'a2');
60 plot(t,x)
61 [x t] = miode(f, [R B C], 0, 140, 1, 10^-6,'a3');
62 plot(t,x)
63 [x t] = miode(f, [R B C], 0, 140, 1, 10^-6,'b1');
64 plot(t,x)
65 [x t] = miode(f, [R B C], 0, 140, 1, 10^-6,'b2');
66 plot(t,x)
67 [x t] = miode(f, [R B C], 0, 140, 1, 10^-6,'b3');
68 plot(t,x)

```

```

69 [x t] = miode(f, [R B C], 0, 140, 1, 10^-6, 'c1');
70 plot(t,x)
71 [x t] = miode(f, [R B C], 0, 140, 1, 10^-6, 'c2');
72 plot(t,x)
73 [x t] = miode(f, [R B C], 0, 140, 1, 10^-6, 'c3');
74 plot(t,x)

```

Listing 4: edos.m

3 RESULTADOS

Aquí se pueden observar gráficamente las soluciones de las EDOs, como se mencionó previamente. Sus nombres $a1, a2, a3, b1, b2, b3, c1, c2, c3$ son consistentes con las indicaciones de los parámetros presentes en la consigna del trabajo práctico.

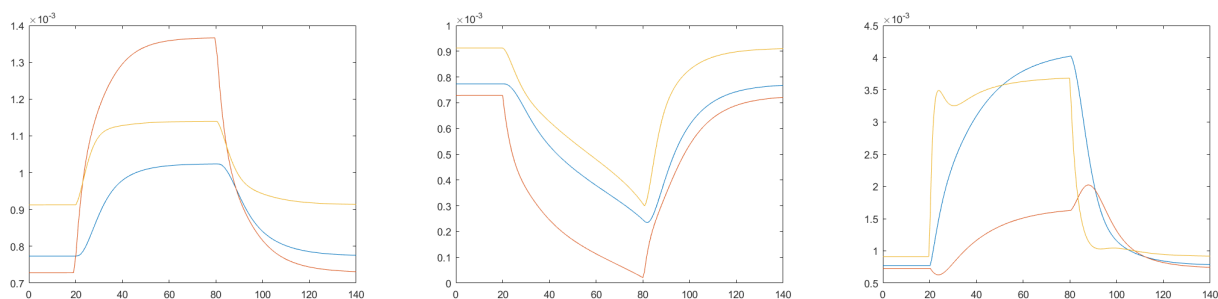


Figure 3.1: Figuras $a1, b1, c1$ respectivamente.

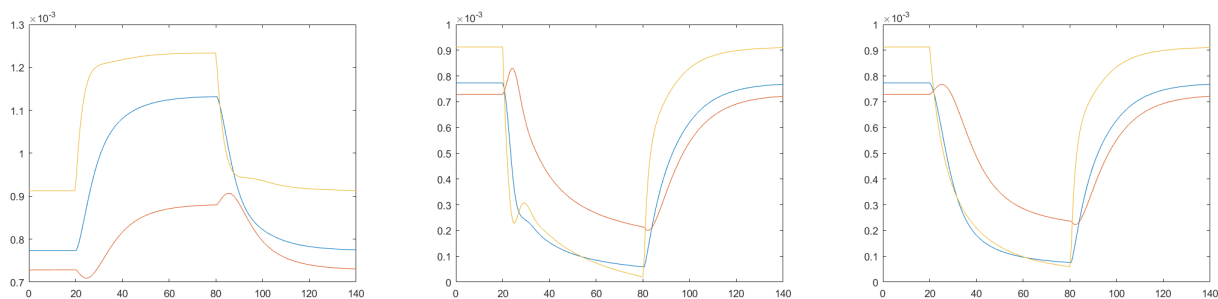


Figure 3.2: Figuras $a2, b2, c2$ respectivamente.

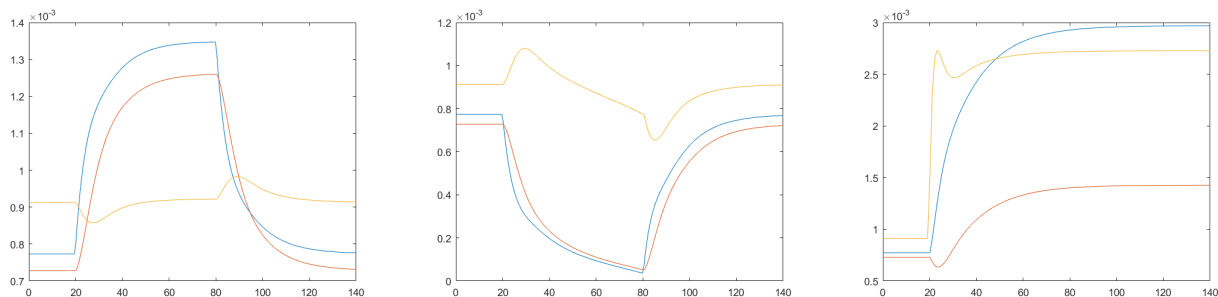


Figure 3.3: Figuras $a3, b3, c3$ respectivamente.

Figure 3.4: Resultados de las EDOs.

Se puede observar que los gráficos se ven como los del Paper mencionado.