6 ECUACIONES DIFERENCIALES ORDINARIAS

Consideremos el problema de valor inicial

$$\frac{dy}{dt} = f(t, y(t)), \quad a \leq t \leq b, \quad y(a) = y_0 \tag{6.1}$$

Buscamos una función $y(t) \in c^1[a,b]$ que satisfaga la identidad (6.1). Se asume que $f(t,y)$ está definida para $t \in [a,b]$ y $y \in \mathbb{R}^m$. Por supuesto, en este texto solo estudiamos el caso $m = 1$.

Existencia y unicidad. En teoría de ecuaciones diferenciales se establece el siguiente teorema,

Teorema 6.1

Si $f(t,y)$ es continua en $t \in [a,b]$ y respecto a y satisface la condición de Lipschitz

$$||f(t,y) - f(t,y^*)|| \leq L||y - y^*||, \quad t \in [a,b], \; y, \, y^* \in \mathbb{R},$$

entonces el problema de valor inicial (6.1) tiene una única solución $y(t)$, $a \leq t \leq b$, para cualquier $y_0 \in \mathbb{R}$.

Supongamos que tenemos el problema de valor inicial

$$\frac{dy}{dt} = f(t,y), \quad a \leq t \leq b, \quad y(a) = y_0 \tag{6.2}$$

Si tenemos una aproximación (t_i, y_i) de $(t_i, y(t_i))$, el paso siguiente en un método de un solo paso es

$$y_{i+1} = y_i + h \cdot \Phi(t_i, y_i; h), \quad h > 0.$$

La función Φ se puede ver como el incremento aproximado en cada paso y define cada método de un solo paso.

Orden del método. Para definir el orden del método necesitamos definir el *error de truncación.* Sea $u(t)$ la solución del problema 6.2 pero pasando por el punto (genérico) (x,y), es decir, $u(t)$ es la solución del problema local

$$\frac{du}{dx} = f(t,u), \quad x \leq t \leq x+h, \quad u(x) = y \tag{6.3}$$

A $u(t)$ se le llama *solución de referencia.* Si $y^* = y + h\Phi(x,y;h)$, y^* aproxima $u(x+h)$ con un *error de truncación* $T(x,y;h) = \dfrac{1}{h}(y^* - u(x+h))$.

Solución numérica. Desde el punto de vista numérico lo que nos interesa encontrar aproximaciones $y_a(t_i)$ a los valores exactos $y(t_i)$. En este capítulo, los $t_i \in [a,b]$ los tomaremos igualmente espaciados, es decir, si $h = (b-a)/n$, $ti = a + i \cdot h$, $i = 0,1,...,n$.

Si uno lo prefiere, puede construir una tabla de aproximaciones $\{(t_i, y_a(ti)),\ i = 0,1,...,n\}$ y por interpolación, contruir una solución aproximada $y_a(t)$, $t \in [a,b]$.

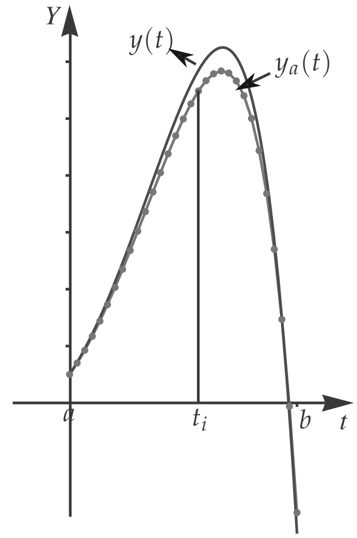


Figura 6.1 Solución numérica de un problema de valor inicial

El método Φ se dice de orden p si $||T(x,y;h)|| \leq Ch^p$ uniformenente sobre $[a,b]$ donde la constante C no depende de x,y o h. Esta propiedad es usual escribirla como

$$T(x,y;h) = O(h^p),\ \ h \to 0,$$

es decir, entre más grande p, más exacto es el método.

A continuación, vamos a ver algunos métodos de un solo paso.

6.1 Método de Euler

Euler propuso este método en 1768. Consiste en seguir la tangente en cada punto (t_i, y_i). Hacemos una partición del intervalo $[a,b]$ en n subintervalos $[t_i, t_{i+1}]$, cada uno de longitud $h = (b-a)/n$. Luego, $t_{i+1} = a + i \cdot h = t_i + h$. Iniciando con (t_0, y_0), se calcula la ecuación de la tangente en (t_i, y_i): $y_T(t) = f(t_i, y_i)(t - t_i) + y_i$ y se evalúa en $t = t_{i+1} = t_i + h$, es decir,

$$y(t_{i+1}) \approx y_{i+1} = y_i + h f(t_i, y_i),\ \ i = 0,1,...,n$$

El método de Euler es de orden $p = 1$.

6.2 Algoritmo e implementación con wxMaxima.

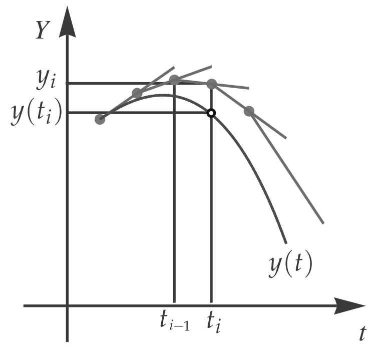


Figura 6.2 $y(t_i) \approx y_i = y_{i-1} + hf(t_{i-1}, y_i)$.

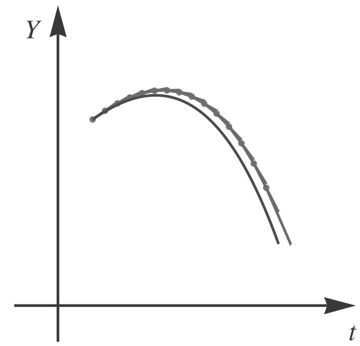


Figura 6.3 Tangentes en (t_i, y_i), $i = 0,1,...15$.

Ejemplo 6.1

Consideremos el problema de valor inicial $\dfrac{dy}{dt} = 0.7y - t^2 + 1$, $t \in [1,2]$, $y(1) = 1$. Aquí $a = 1, b = 2$. Si $n = 10$ entonces $h = 0.1$ y $t_i = a + hi = 1 + 0.1i$

$$\begin{cases} y_0 &= 1 \\ y_{i+1} &= y_i + h(0.7 * y_i - t_i{}^2 + 1) = y_i + 0.1\left(0.7y_i - (1 + 0.1i)^2 + 1\right), \quad i = 1, ..., n \end{cases}$$

i	t_i	y_i
0	1	1
1	1.1	1.07
2	1.2	1.1239
3	1.3	1.15857
4	1.4	1.17067
5	1.5	1.15662
6	1.6	1.11258
7	1.7	1.03446
8	1.8	0.917877
9	1.9	0.758128
10	2.	0.550197

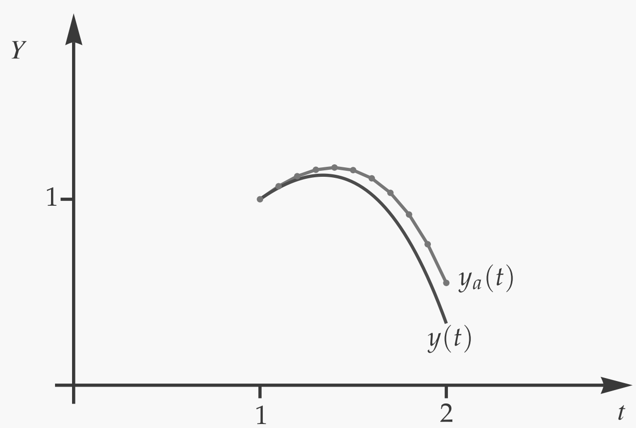


Tabla 6.1 $y(t_i) \approx y_i = y_{i-1} + hf(t_{i-1}, y_i)$, $i = 1, 2, ..., 10$.

En el algoritmo se usan las variables t_0, y_0 y y_1. En cada ciclo t_0 y y_0 se actualizan para calcular la siguiente aproximación $y_1 = f(t_0, y_0) \cdot h + y0$. Luego se incrementa t_0 y se actualiza $y_0 = y_1$.

Algoritmo 6.1: Método de Euler

Datos: $f(t,y)$, a, b, y_0, n

Salida: Imprime las aproximaciones (t_i, y_i), $i = 0,1,...,n$

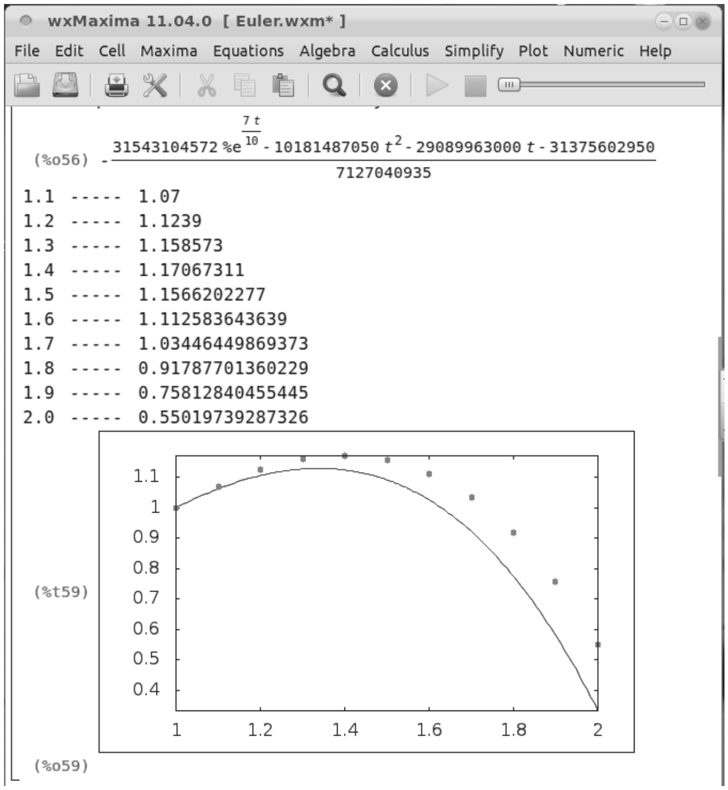
$h = (b-a)/n;$
$t_0 = a;$
for $i = 1$ **to** n **do**
  $y_1 = y_0 + h \cdot f(t_0, y_0);$
  $t_0 = a + i \cdot h;$
  $y_0 = y_1;$
  print((t_0, y_0));

Implementación en wxMaxima. En esta primera implementación, definimos la función $f(t,y)$ y luego la llamamos en el programa por su nombre `'f'`.

```
f(t,y):=0.7*y-t^2+1;  /*dy/dt = f(t,y)*/
Euler(f,a,b,y00,n):=block([h:(b-a)/n*1.0,y0:y00,y1,t0],
print(a,"------",y0*1.0),
t0:a,
for  i:1 thru n do (
        y1:y0+h*f(t0,y0),
        /*Actualizamos t0 y y0*/
        t0:a+i*h,
        y0:y1,
        print(t0,"------",y0*1.0)
                   )
)%$
/*Llamada al programa*/
Euler(f,1,2,1,10);
```

```
/* Salida */
1 ------ 1.0
1.1 ------ 1.07
1.2 ------ 1.1239
1.3 ------ 1.158573
1.4 ------ 1.17067311
1.5 ------ 1.1566202277
1.6 ------ 1.112583643639
1.7 ------ 1.03446449869373
1.8 ------ 0.91787701360229
1.9 ------ 0.75812840455445
2.0 ------ 0.55019739287326
```

Segunda implementación. Esta segunda versión tiene fines didácticos. Se acepta la expresión `'fty'` sin importar el nombre de las variables. Si las variables son `'t'` y `'y'`, en ese orden,debemos hacer la sustitución de variables, usando `'substitute'` y luego definir la función `'f(t,y)'`. Se usa el paquete `'draw2d'` para desplegar los gráficos: Una lista con los puntos (t_{i+1}, y_{i+1}) y la solución exacta (explícita) $y = y(t)$ de la ecuación diferencial. La salida esperada del programa será



Si la ecuación diferencial tiene solución implícita entonces se debe cambiar el código en la parte del cálculo de la solución de la ecuación diferencial y el gráfico, porque se debe graficar ahora una solución implícita. Por ejemplo, la ecuación diferencial $\dfrac{dy}{dt} = t/y$ con $y(2) = -1$ tiene solución implícita $\dfrac{y^2}{2} = \dfrac{t^2-3}{2}$. Habría que hacer el siguiente ajuste,

Por supuesto, este programa sirve también para el caso explícito.

6.3 Métodos de Taylor de orden superior.

El método de Euler opera con un polinomio de Taylor de orden uno (rectas). Es natural, como propuso Euler, usar más términos e la expansión de Taylor (si f es suficientemente derivable). Usando una expansión de orden m nos lleva a un método de orden $O(h^m)$. El costo de calcular las derivadas en estos tiempos se le delega a los computadores, lo que hace que el método (todavía de un solo paso), sea una opción viable.

En este método, calculamos el polinomio de Taylor alrededor de $t = t_i$ (en potencias de $t - t_i$) y evaluamos este polinomio en $t_{i+1} = t_i + h$. Nos queda un polinomio en potencias de h,

$$y(t_{i+1}) = y(t_i+h) = y(t_i) + hy'(t_i) + \frac{h^2}{2}y''(t_i) + ... + \frac{h^m}{m!}y^{(m)}(t_i) + \frac{h^{m+1}}{(m+1)!}y^{(m+1)}(\xi_i), \quad \text{con} \quad \xi_i \in]t_i,\, t_i+h[.$$

Como $y'(t) = f(t,y)$, las derivadas sucesivas se pueden calcular usando regla de la cadena (en dos variables).

$$\begin{cases} y^{(0)}(t) &= f^{[0]}(t,y) = f(t,y), \\ y^{(k+1)}(t) &= f^{[k+1]}(t,y) = f_t^{[k]}(t,y) + f_y^{[k]}(t,y)\,f(t,y), \quad k = 0,1,2,...,m. \end{cases}$$

Entonces, sacando h a factor,

$$y(t_{i+1}) \approx y_{i+1} = y_i + h\left[f^{[0]}(t_i,y_i) + \frac{h}{2}f^{[1]}(t_i,y_i) + +\frac{h^2}{3!}f^{[2]}(t_i,y_i) + ... + \frac{h^{m-1}}{m!}f^{[m-1]}(t_i,y_i)\right],\ i = 0,1,...,n. \tag{6.4}$$

Ejemplo 6.2

Consideremos el problema de valor inicial $\dfrac{dy}{dt} = 0.7y - t^2 + 1$, $t \in [1,2]$, $\quad y(1) = 1$. La solución exacta es $y(t) = 1.42857t^2 + 4.08163t - 4.42583e^{0.7t} + 4.40233$.

Vamos a aplicar el método de Taylor de orden $m = 4$ con $n = 10$. Tenemos $a = 1$, $b = 2$, $h = 0.1$ y $t_i = 1 + 0.1i$. Ahora debemos calcular las derivadas,

Ejemplo 6.2 (continuación).

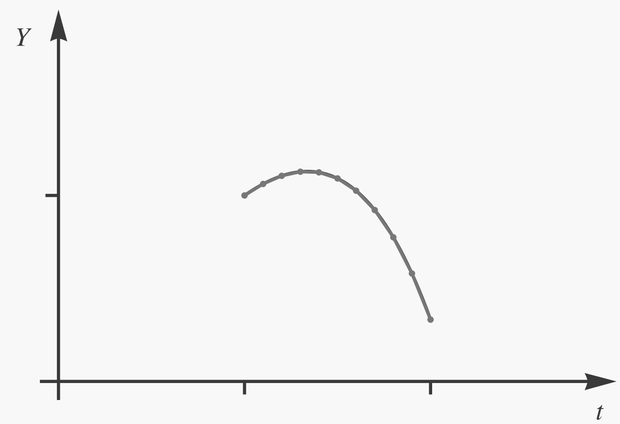
$$\begin{aligned} f^{[0]}(t,y) &= 0.7y - t^2 + 1, \\ f^{[1]}(t,y) &= -2t + 0.7(0.7y - t^2 + 1), \\ f^{[2]}(t,y) &= -2 - 2\cdot 0.7^1 t + 0.7^2(0.7y - t^2 + 1), \\ f^{[3]}(t,y) &= -2\cdot 0.7^1 - 2\cdot 0.7^2 t + 0.7^3(0.7y - t^2 + 1). \end{aligned}$$

$$\begin{cases} y_0 &= 1 \\ y_{i+1} &= y_i + h\left[f^{[0]}(t_i,y_i) + \frac{h}{2}f^{[1]}(t_i,y_i) + \frac{h^2}{3!}f^{[2]}(t_i,y_i) + \frac{h^3}{4!}f^{[3]}(t_i,y_i)\right],\ i = 0,1,...,n. \end{cases}$$

Así, $y_1 = y_0 + 0.1\left[f^{[0]}(t_0,y_0) + \frac{0.1}{2}f^{[1]}(t_0,y_0) + \frac{0.1^2}{3!}f^{[2]}(t_0,y_0) + \frac{0.1^3}{4!}f^{[3]}(t_0,y_0)\right] = 1.06193158375.$

En la tabla se continúa el cálculo hasta y_{10}.

t_i	Método de Taylor de orden 4 y_i	Valor exacto $y(t_i)$
1.0	1.0	1.0
1.1	1.06193158375	1.061931432036279
1.2	1.105577521330614	1.105577223160223
1.3	1.127540292137498	1.127539827069448
1.4	1.124176027574661	1.124175372973675
1.5	1.091576648813122	1.091575779638891
1.6	1.025550709391196	1.025549597968614
1.7	0.92160284874683	0.92160146451555
1.8	0.77491175596324	0.77491006520478
1.9	0.58030653570613	0.58030450124654
2.0	0.33224136049833	0.33223894138437



6.4 Algoritmo e implementación con wxMaxima.

Para el algoritmo vamos a reescribir la relación 6.4 como $y_{i+1} = yi + T^{[m]}(t_i,y_i)$ donde,

$$T^{[m]}(t,y) = h\,f^{[0]}(t_i,y_i) + \frac{h^2}{2}f^{[2]}(t_i,y_i) + ... + \frac{h^m}{m!}f^{[m-1]}(t_i,y_i).$$

En el algoritmo se usan las variables t_0, y_0 y y_1. En cada ciclo t_0 y y_0 se actualizan para calcular la siguiente aproximación $y_1 = T^{[}k](t_0, y_0)$. Luego se incrementa t_0 y se actualiza $y_0 = y_1$.

Algoritmo 6.2: Método de Taylor de orden m

Datos: $f(t,y)$, m, a, b, y_0, n
Salida: Imprime las aproximaciones (t_i, y_i), $i = 0, 1, ..., n$

$h = (b-a)/n$;
$t_0 = a$;
for $i = 1$ **to** n **do**
  $y_1 = y_0 + T^{[m]}(t_0, y_0)$;
  $t_0 = a + i \cdot h$;
  $y_0 = y_1$;
  print((t_0, y_0));

Implementación en wxMaxima. En esta primera implementación, definimos la función $f(t,y)$ y luego la llamamos en el programa por su nombre `'f'`. Por supuesto, la derivadas parciales se calculan con `fk:f(t,y)` y `fk:diff(fk,t) + diff(fk,y)*f(t,y)`,

```
/* Salida */
1.1  ------  1.06193158375
1.2  ------  1.105577521330614
1.3  ------  1.127540292137498
1.4  ------  1.124176027574661
1.5  ------  1.091576648813122
1.6  ------  1.025550709391196
1.7  ------  0.92160284874683
1.8  ------  0.77491175596324
1.9  ------  0.58030653570613
2.0  ------  0.33224136049833
```

Para agregar el gráfico en el programa, para propósitos didácticos, debemos agregar cada nuevo punto `(t0,y0)` en una matriz. Aquí calculamos la solución exacta `yt` de la ecuación diferencial (si hubiera) y usamos `implicit(yt,t,a,b,y,-3,3)` para graficar. Se imprime los valores exactos $y(t_i)$. En vez de `-3,3` se podría usar $[\text{mín}\{yi\} - 1, \text{máx}\{yi\} + 1]$.

6.5 Métodos de Runge-Kutta.

Como decíamos, los métodos de un solo paso tienen la forma

$$y_{i+1} = y_i + h \cdot \Phi(t_i, y_i; h), \quad h > 0.$$

En el método de Euler la *función incremento* es

$$\Phi(t_i, y_i; h) = f(t_i, y_i)$$

Para el método de Taylor de orden 2 es,

$$\Phi(t_i, y_i; h) = f(t_i, y_i) + \frac{h}{2}\left[f_t(t_i, y_i) + f_y(t_i, y_i) f(t_i, y_i)\right] \tag{6.5}$$

Los métodos de Runge-Kutta son métodos diseñados pensando en imitar las expansiones de Taylor pero usando solo evaluaciones de la función $f(t, y)$. En el caso del método de Runge-Kutta de orden 2, se trata de modificar el método de Euler escribiendo

$$\Phi(t_i, y_i; h) = a_1 k_1 + a_2 k_2, \tag{6.6}$$

con $k_1 = f(t, y)$ y $k_2 = f(t + \alpha h,\ y + \beta h k_1)$, es decir, no se va a evaluar en la tangente hasta $t_i + h$ sino antes, usando la pendiente de la tangente en $(t + \alpha h,\ y + \beta h k_1)$.

Expandiendo 6.5 y 6.6 en potencias de h (usando la fórmula de Taylor en dos variables) y comparando se obtiene, entre varias opciones, $\alpha = 1$, $\beta = 1$, $a_1 = a_2 = 1/2$. Esto nos da un método Runge-Kutta de orden 2,

$$y_{i+1} = y_i + \frac{h}{2}\left[f(t_i, y_i) + f(t_i + h,\ y_i + h f(t_i, y_i))\right]$$

El método clásico de Runge-Kutta de orden 4 tiene una función de incremento que coincide con el polinomio de Taylor hasta el sumando con el término h^4. Este método se puede escribir como,

$$\begin{cases} y_0 &= y(a), \\ y_{i+1} &= y_i + \dfrac{1}{3}(k_1 + 2k_2 + 2k_3 + k_4),\ \ i = 0, 1, 2, \ldots \end{cases}$$

donde

$$\begin{aligned} k_1 &= \frac{h}{2} f(t_i, y_i), \\ k_2 &= \frac{h}{2} f\left(t_i + \frac{h}{2},\ y_i + k_1\right), \\ k_3 &= \frac{h}{2} f\left(t_i + \frac{h}{2},\ y_i + k_2\right), \\ k_4 &= \frac{h}{2} f\left(t_i + h,\ y_i + 2k_3\right), \end{aligned}$$

6.6 Algoritmo e implementación con WXMAXIMA.

Algoritmo 6.3: Método de Runge-Kutta de orden 4

Datos: $f(t,y)$, m, a, b, y_0, n

Salida: Imprime las aproximaciones (t_i,y_i), $i=0,1,...,n$

$h=(b-a)/n$;

$t_0=a$;

for $i=1$ **to** n **do**

$\quad k_1=\frac{h}{2}f(t_i,y_i)$;

$\quad k_2=\frac{h}{2}f\left(t_i+\frac{h}{2},\, y_i+k_1\right)$;

$\quad k_3=\frac{h}{2}f\left(t_i+\frac{h}{2},\, y_i+k_2\right)$;

$\quad k_4=\frac{h}{2}f\left(t_i+h,\, y_i+2k_3\right)$;

$\quad y_1=y_i+\frac{1}{3}(k_1+2k_2+2k_3+k_4)$;

$\quad t_0=a+i\cdot h$;

$\quad y_0=y_1$;

$\quad$ print$((t_0,y_0))$;

Implementación en WXMAXIMA.

EJERCICIOS

6.1 Considere el problema de valor inicial $y'=\cos(2t)+\text{sen}(3t)$, $t\in[0,1]$, $y(0)=1$.

a) Usando el método de Euler, aproximar $y(0.4)$ con $h=0.1$.

b) Usando el método de Taylor de orden 4, aproximar $y(0.4)$ con $h=0.2$

c) Usando el método de Runge-Kutta de orden 4, aproximar $y(0.4)$ con $h=0.2$

6.2 Considere el problema de valor inicial $y'=t\,\text{Exp}(3t)-40y$, $t\in[1,2]$, $y(1)=10$.

a) Usando el método de Euler, aproximar $y(0.4)$ con $h=0.1$.

b) Usando el método de Taylor de orden 4, aproximar $y(0.4)$ con $h=0.2$

c) Usando el método de Runge-Kutta de orden 4, aproximar $y(0.4)$ con $h=0.2$

6.3 Usando el método de Runge-Kutta de orden 4, aproximar $y(0.2)$ (con $h=0.1$) si $y(t)=\int_0^t e^{-t^2}\,dt$. (Debe convertir el cálculo de la integral en un problema de valor inicial.)

6.7 Algunos Detalles Teóricos.

Definición 6.1

Consideremos un conjunto $D \subseteq \mathbb{R}^2$ y una función $f(t,y)$ definida en D. Si existe una constante $L > 0$ tal que

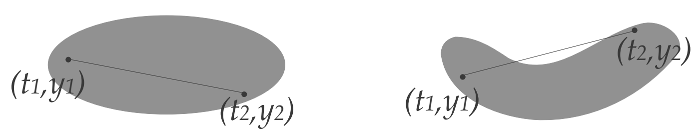
$$|f(t,y_1) - f(t,y_2)| \leq L|y_1 - y_2|, \quad \forall (t,y_1), (t,y_2) \in D$$

se dice que $f(t,y)$ cumple una condición de Lipschitz en la variable y en D. A L se le llama constante de Lipschitz para f.

Nota: Una condición *suficiente* para que $f(t,y)$ cumpla una condición de Lipschitz en D es que exista $L > 0$ tal que $\left|\frac{\partial f}{\partial y}(t,y)\right| \leq L, \quad \forall (t,y) \in D$

Definición 6.2

Un conjunto $D \subseteq \mathbb{R}$ se dice convexo si $\forall (t_1,y_1),(t_2,y_2) \in D$, el segmento $\{(1-\lambda)(t_1,y_1) + \lambda(t_2,y_2), \quad \lambda \in [0,1]\}$ está contenido en D.



Convexo *No convexo*

Nota: Observe que, cuando $\lambda = 0$ estamos en el punto inicial (t_1,y_1) y cuando $\lambda = 1$ estamos en el punto final (t_2,y_2). $\lambda = 1/2$ corresponde al punto medio del segmento.

Teorema 6.2

Si $D = \{(t,y) : a \leq t \leq b, -\infty \leq y \leq \infty\}$ y si $f(t,y)$ es continua en D y satisface una condición de Lipschitz respecto a y en D, entonces el problema $(*)$ tiene una solución única $y(t)$ para $a \leq t \leq b$.

Nota: Un problema de valor inicial está *bien planteado* si pequeños cambios o perturbaciones en el planteo del problema (debido a errores de redondeo en el problema inicial, por ejemplo), ocasiona cambios pequeños en la solución del problema. Si un problema cumple las hipótesis del teorema anterior, entonces está bien planteado.

Ejemplo 6.3

Consideremos el problema de valor inicial $y' = y - t^2 + 1,\ t \in [0,4],\quad y(0) = 0.5$. Aquí $f(t,y) = y - t^2 + 1$. Si $D = \{(t,y) : 0 \leq t \leq 4, -\infty \leq y \leq \infty\}$, entonces como

$$\left|\frac{\partial f}{\partial y}(t,y)\right| = |1| \quad \forall\,(t,y) \in D$$

f cumple una condición de Lipschitz en y (en este caso podemos tomar $L = 1$). Además, como $f(t,y)$ es continua en D, el problema de valor inicial tiene una solución única. De hecho la única solución es $y(t) = (t+1)^2 - 0.5e^t$

6.8 Estimación del error

Teorema 6.3

Si $D = \{(t,y) : a \leq t \leq b, -\infty \leq y \leq \infty\}$ y si $f(t,y)$ es continua en D y satisface una condición de Lipschitz respecto a y en D con constante L entonces si existe una constante M tal que

$$|y''(t)| \leq M,\ \ \forall\, t \in [a,b]$$

entonces para cada $i = 0,1,2,...,n,$

$$|y(t_i) - y_i| \leq \frac{hM}{2L}\left(e^{L(t_i-a)} - 1\right)$$

Nota: para calcular $|y''(t)|$ usamos regla de la cadena: $y''(t) = \dfrac{\partial f}{\partial t}(t,y) + \dfrac{\partial f}{\partial y}(t,y) \cdot y'(t)$. Posiblemente sea difícil obtener M dado que puede ser necesaria información acerca de $y(t)$.

Teorema 6.4 (Teorema de Taylor).

Sea $h > 0$ y f una función tal que f y sus primeras k derivadas son continuas en el intervalo $[a, a+h]$ y la derivada f^{k+1} existe en $]a, a+h[$, entonces existe un número ξ , en el intervalo $]a, a+h[$ tal que

$$f(a+h) = f(a) + \frac{f'(a)}{1!}h + \frac{f''(a)}{2!}h^2 + \cdots + \frac{f^{(k)}(a)}{k!}h^k + \frac{f^{(k+1)}(\xi)}{(k+1)!}h^{k+1}$$