Método del disparo para EDOs de 2do orden con valores en la frontera

Proyecto - Métodos Numéricos 2023

Miguel Angel Ruiz Ortiz

29 de noviembre del 2023

1 Introducción

En este trabajo se estudia el **método del disparo** para resolver ecuaciones diferenciales de segundo orden con valores en la frontera, es decir, encontrar la solución al problema

$$u'' = f(x, u, u'), \quad x \in [a, b], \quad u(a) = \alpha,\ u(b) = \beta, \tag{1}$$

donde las condiciones $u(a) = \alpha$ y $u(b) = \beta$ se conocen como *condiciones de frontera*. En esta notación, las derivadas son respecto a x.

A diferencia del problema del valor inicial de orden 2, en el que se resuelve una ecuación diferencial de 2do orden pero con las condiciones $u(a) = \alpha_1$ y $u'(a) = \alpha_2$ especificadas en el punto inical a, los problemas con valores en la frontera tienen condiciones impuestas para u en los extremos a y b. Ejemplos de este tipo de problemas se encuentran en la física y la ingeniería.

La idea del método es reducir dicho problema al problema del valor inicial (PVI). En general, el método del disparo es un método iterativo en el que se aproxima la solución $u(x)$ al problema de valores en la frontera (1) mediante una sucesión de soluciones $u_k(x)$ a los problemas del valor inicial

$$u'' = f(x, u, u'), \quad x \in [a, b], \quad u(a) = \alpha,\ u(b) = t_k, \tag{2}$$

parametrizados por una sucesión $\{t_k\}_k$ tal que

$$\lim_{k\to\infty} u_k(b) = u(b) = \beta.$$

Sin embargo, en el caso del problema *lineal* veremos cómo la solución es una combinación lineal de soluciones a dos problemas del valor inicial, por lo que se obtiene un método directo.

El siguiente teorema da condiciones generales para asegurar que existe una única solución al problema de valores de la frontera (1). Su demostración se encuentra en Keller (1968, Teorema 1.2.2).

Teorema 1.1 *Sea $D = [a, b] \times \mathbb{R} \times \mathbb{R} \subset \mathbb{R}^3$ y $f : D \to \mathbb{R}$ una función continua con derivadas parciales $\frac{\partial f}{\partial y}$ y $\frac{\partial f}{\partial z}$ también continuas. Si*

- $\frac{\partial f}{\partial y}(x,y,z) > 0$ *para todo* $(x,y,z) \in D$, *y*
- *existe* $M > 0$ *tal que*
$$\left| \frac{\partial f}{\partial z}(x,y,z) \right| \leq M$$
para todo $(x,y,z) \in D$,

entonces el problema con valores en la frontera

$$u'' = f(x,u,u'), \quad x \in [a,b], \quad u(a) = \alpha,\ u(b) = \beta,$$

tiene una única solución.

■

2 Problema lineal con valores en la frontera

Se dice que una ecuación diferencial de 2do orden es *lineal* si es de la forma

$$u'' = p(x)u' + q(x)u + r(x),$$

para ciertas funciones $p(x), q(x)$ y $r(x)$. En este caso, el Teorema 1.1 puede ser simplificado con el siguiente corolario.

Corolario 2.1 *Supongamos que el problema lineal con valores en la frontera*

$$u'' = p(x)u' + q(x)u + r(x), \quad x \in [a,b], \quad u(a) = \alpha,\ u(b) = \beta, \tag{3}$$

satisface que $p(x), q(x)$ *y* $r(x)$ *son continuas en* $[a,b]$ *y* $q(x) > 0$ *en* $[a,b]$. *Entonces el problema tiene una única solución.*

Demostración. Sea $f(x,y,z) = p(x)z + q(x)y + r(x)$. Por ser p, q y r continuas en $[a,b]$, entonces f es continua en $D := [a,b] \times \mathbb{R} \times \mathbb{R}$. Además,

$$\frac{\partial f}{\partial y}(x,y,z) = q(x), \quad \frac{\partial f}{\partial z}(x,y,z) = p(x)$$

son continuas en D,

- $\frac{\partial f}{\partial y}(x,y,z) = q(x) > 0$ para todo $(x,y,z) \in D$, y
- $\left| \frac{\partial f}{\partial z}(x,y,z) \right| = |p(x)| \leq M := \sup_{x \in [a,b]} |p(x)|$, para todo $(x,y,z) \in D$, donde $M < \infty$ ya que p es continua en $[a,b]$ y toda función continua en un intervalo cerrado es acotada.

Dado que el problema (3) se escribe como $u'' = f(x,u,u')$ y f satisface las condiciones del Teorema 1.1, se sigue que existe una única solución al problema deseado.

■

2.1 Método del disparo: caso lineal

Supongamos $p(x), q(x)$ y $r(x)$ son continuas en $[a, b]$, con $q(x) > 0$, y sea $D := [a, b] \times \mathbb{R} \times \mathbb{R}$. Veamos cómo construir una solución u al problema lineal (3). Consideremos los siguientes problemas del valor inicial

$$u'' = p(x)u' + q(x)u + r(x), \quad a \leq x \leq b, \quad u(a) = \alpha,\ u'(a) = 0, \tag{4}$$

y

$$u'' = p(x)u' + q(x)u, \quad a \leq x \leq b, \quad u(a) = 0,\ u'(a) = 1. \tag{5}$$

Estas ecuaciones de 2do orden son equivalentes a los siguientes sistemas de ecuaciones de 1er orden, respectivamente:

$$\begin{aligned} u' &= v \\ v' &= p(x)v + q(x)u + r(x), \quad a \leq x \leq b, \quad u(a) = \alpha,\ v(a) = 0, \end{aligned} \tag{6}$$

y

$$\begin{aligned} u' &= v \\ v' &= p(x)v + q(x)u, \quad a \leq x \leq b, \quad u(a) = 0,\ v(a) = 1. \end{aligned} \tag{7}$$

Recordemos que estos sistemas tienen solución única si las funciones que los definen $F, G : D \to \mathbb{R}^2$, dadas como $F(x, u, v) = (v, p(x)v + q(x)u + r(x))$ y $G(x, u, v) = (v, p(x)v + q(x)u)$, son continuas en D y existen $L_F, L_G > 0$ tales que

$$\begin{aligned} ||F(x, u_1, v_1) - F(x, u_2, v_2)||_2 &\leq L_F ||(u_1, v_1) - (u_2, v_2)||_2, \\ ||G(x, u_1, v_1) - G(x, u_2, v_2)||_2 &\leq L_G ||(u_1, v_1) - (u_2, v_2)||_2, \end{aligned}$$

para todo $(x, u_1, v_1), (x, u_2, v_2) \in D$, i.e., son Lipschitz en las variables u, v. En efecto, por ser p y q continuas en el intervalo cerrado $[a, b]$ podemos tomar

$$C := \max \left\{ \sup_{x \in [a,b]} |p(x)|, \sup_{x \in [a,b]} |q(x)| \right\} < \infty,$$

y obtenemos las siguientes desigualdades:

$$\begin{aligned} &||G(x, u_1, v_1) - G(x, u_2, v_2)||_2^2 = ||F(x, u_1, v_1) - F(x, u_2, v_2)||_2^2 = ||(v_1 - v_2, p(x)(v_1 - v_2) + q(x)(u_1 - u_2)||_2^2 \\ &= |v_1 - v_2|^2 + |p(x)(v_1 - v_2) + q(x)(u_1 - u_2)|^2 \leq |v_1 - v_2|^2 + (|p(x)||v_1 - v_2| + |q(x)||u_1 - u_2|)^2 \\ &\leq |v_1 - v_2|^2 + C^2(|v_1 - v_2| + |u_1 - u_2|)^2 \leq |v_1 - v_2|^2 + C^2(2|v_1 - v_2|^2 + 2|u_1 - u_2|^2) \quad (\star) \\ &\leq (2C^2 + 1)(|v_1 - v_2|^2 + |u_1 - u_2|^2) = (2C^2 + 1)||(u_1, v_1) - (u_2, v_2)||_2^2. \end{aligned}$$

La desigualdad $(\star)$ es debido a que $2|a||b| \leq |a|^2 + |b|^2$ para todo $a, b \in \mathbb{R}$ (tomar $a = v_1 - v_2$ y $b = u_1 - u_2$). De lo anterior se sigue que

$$||G(x, u_1, v_1) - G(x, u_2, v_2)||_2 = ||F(x, u_1, v_1) - F(x, u_2, v_2)||_2 \leq \sqrt{(2C^2 + 1)}||(u_1, v_1) - (u_2, v_2)||_2,$$

para todo $(x, u_1, v_1), (x, u_2, v_2) \in D$. Por lo que los sistemas (6) y (7) tienen solución única.

Estos sistemas se pueden resolver mediante el método de *Runge-Kutta* (de cuarto orden) (ver Apéndice A). Con dicho método obtenemos soluciones u_1 y u_2 a los problemas del valor inicial (4) y (5), respectivamente. Se puede demostrar que $u_2(b) \neq 0$ bajo las condiciones del Corolario 2.1 (si $u_2(b) = 0$, se demuestra que $u_2 = 0$, por lo que $u_2'(a) = 0$ y u_2 ya no cumpliría las condiciones iniciales del PVI (5)). Si definimos

$$u(x) = u_1(x) + \frac{\beta - u_1(b)}{u_2(b)} u_2(x), \tag{8}$$

entonces u es solución del problema con valores en la frontera lineal (3):

$$\begin{aligned}
u'' &= u_1'' + \frac{\beta - u_1(b)}{u_2(b)} u_2'' = (p(x)u_1' + q(x)u_1 + r(x)) + \frac{\beta - u_1(b)}{u_2(b)} (p(x)u_2' + q(x)u_2) \\
&= p(x) \left(u_1' + \frac{\beta - u_1(b)}{u_2(b)} u_2' \right) + q(x) \left(u_1 + \frac{\beta - u_1(b)}{u_2(b)} u_2 \right) + r(x) \\
&= p(x)u' + q(x)u + r(x), \\
u(a) &= u_1(a) + \frac{\beta - u_1(b)}{u_2(b)} u_2(a) = \alpha, \quad u(b) = u_1(b) + \frac{\beta - u_1(b)}{u_2(b)} u_2(b) = \beta.
\end{aligned}$$

Este proceso define el algoritmo para el método del disparo en el caso lineal.

2.1.1 Algoritmo

El algoritmo del método del disparo para el caso lineal recibe las funciones $p(x)$, $q(x)$, $r(x)$ que definen el problema, los extremos del intervalo a y b, y las condiciones de frontera α y β en a y b, respectivamente. También recibe un entero positivo n que es el número de divisiones que se harán del intervalo $[a, b]$, de tal manera que el algoritmo regresa la función solución u evaluada en $x_i := a + i \cdot \frac{b-a}{n}$, $i = 0, \ldots, n$, como vector $(u(x_0), \ldots, u(x_n))$. Dentro del algoritmo se manda a llamar una función `RungeKutta` que implementa dicho método. La función `RungeKutta`, al resolver un sistema de orden m, regresa una matriz $U = (u_{ij})$ de $(n+1) \times m$ tal que u_{ij} es la j-ésima solución del sistema evaluada en el punto x_i, $j = 0, \ldots, m-1$, $i = 0, \ldots, n$ (ver Apéndice A para ver su implementación).

En el notebook de Python adjunto se muestra el código de este algoritmo y se resuelve un ejemplo.

---

function MetodoDisparoLineal($p(x), q(x), r(x), a, b, \alpha, \beta, n$)

$F(x, u, v) \leftarrow (v, p(x)v + q(x)u + r(x))$ // función F que define el sistema (6)
conditions_F $\leftarrow$ (α, 0) // $u_1(a)$ y $u_1'(a)$, resp.

$G(x, u, v) \leftarrow (v, p(x)v + q(x)u)$ // función G que define el sistema (7)
conditions_G $\leftarrow$ (0, 1) // $u_2(a)$ y $u_2'(a)$, resp.

// Resolver sistema (6):
$U \leftarrow$ `RungeKutta`(F, conditions_F, a, b, n) // la columna 0 de U contiene la solución u_1 de (4).

// Resolver sistema (7):
$V \leftarrow$ `RungeKutta`(G, conditions_G, a, b, n) // la columna 0 de V contiene la solución u_2 de (5).

$u \leftarrow (0, \ldots, 0)$ // inicializar vector con $n + 1$ ceros.
for $i \in [0, \ldots, n]$ **do**

$$u_i = U_{i,0} + \frac{\beta - U_{n,0}}{V_{n,0}} V_{i,0}$$

end for

return u
end function

2.1.2 Análisis del error

Recordemos que el método de Runge-Kutta (de orden 4) tiene un error del orden $O(h^4)$, donde $h = (b-a)/n$. Dado que la solución u al problema lineal con valores en la frontera es una combinación lineal de dos soluciones u_1 y u_2 a sistemas de problemas de valor inicial que se resuelven con Runge-Kutta, se sigue que el método del disparo, en el caso lineal, tiene error del orden $O(h^4)$ también. En caso de resolver los sistemas con algún otro método de orden $O(h^k)$, obtendremos que el método del disparo tendrá error $O(h^k)$:

$$|u(x_i) - \hat{u}_i| \leq Kh^k \left|1 + \frac{\hat{u}_{2,i}}{\hat{u}_{2,n}}\right|,$$

donde K es alguna constante positiva, $x_i := a + ih$, $\hat{u}_i$ es la aproximación a $u(x_i)$ dada por el método, y $\hat{u}_{2,i}, \hat{u}_{2,n}$ son las aproximaciones obtenidas de $u_2(x_i)$ y $u_2(x_n) = u_2(b)$ respectivamente.

Lo anterior es suponiendo que no hay errores de redondeo. Sin embargo, puede haber problemas de errores de redondeo al calcular la combinación lineal de las soluciones u_1 y u_2. Recordemos la solución al problema lineal con valores en la frontera era de la forma

$$u(x) = u_1(x) + \frac{\beta - u_1(b)}{u_2(b)} u_2(x).$$

Cuando u_1 crece mucho en el intervalo $[a, b]$, entonces $u_1(b)$ será grande. En este caso, si β es pequeño en comparación de $u_1(b)$, entonces el cociente $\frac{\beta - u_1(b)}{u_2(b)}$ será aproximadamente $-\frac{u_1(b)}{u_2(b)}$, y podríamos perder precisión.

3 Problema no lineal con valores en la frontera

3.1 Método del disparo: caso no lineal

En el caso general cuando se quiere resolver el problema con valores en la frontera

$$u'' = f(x, u, u'), \quad x \in [a, b], \quad u(a) = \alpha,\ u(b) = \beta, \tag{9}$$

y f no tiene la forma como en el caso lineal estudiado en la sección anterior, aproximaremos la solución a través de una secuencia de soluciones de problemas de valor inicial parametrizados con una variable t:

$$u'' = f(x, u, u'), \quad x \in [a, b], \quad u(a) = \alpha,\ u'(a) = t. \tag{10}$$

Se escogen parámetros $t = t_k$ tales que

$$\lim_{k\to\infty} u_k(b) = u(b) = \beta, \tag{11}$$

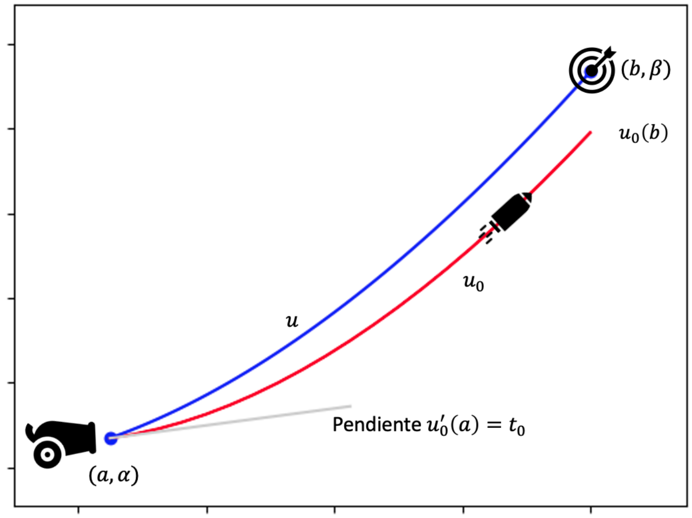

Figura 1: Se llama "método del disparo" porque el método se asemeja a "disparar" varias veces con cierta pendiente t_k desde el punto (a, α) queriendo dar al objetivo (b, β). La bala del disparo tendría la trayectoria dada por la solución u_k de (10) obtenida. En cada disparo corregimos la pendiente t_k de tal manera que el disparo esté más cerca al objetivo.

donde u_k es la solución a (10) con $t = t_k$, y la solución al problema con valores en la frontera (9) sería

$$u(x) := \lim_{k\to\infty} u_k(x, t_k).$$

Supongamos que f satisface las hipótesis del Teorema 1.1 y denotemos por $u(x)$ la solución del problema con valores en la frontera (9). Denotemos también por $u(x,t)$ a la solución del problema del valor inicial (10). Empezamos con un parámetro inicial t_0 y encontramos $u(x, t_0)$. Si $u(b, t_0)$ no está suficientemente cerca de β, corregimos nuestra aproximación escogiendo t_1, y así sucesivamente hasta que $u(b, t_k)$ esté suficientemente cerca de β (ver Figura 1).

Para determinar los parámetros t_k, necesitamos considerar la ecuación en t

$$u(b,t) - \beta = 0, \tag{12}$$

y podemos aproximar su raíz mediante el método de la secante o el método de Newton. Notar una raíz de la ecuación (12) determina una solución al problema con valores en la frontera (9) y también una solución al problema (9) determina una raíz para esta ecuación. De manera que bajo las hipótesis del Teorema 1.1, la ecuación (12) tiene una única raíz.

Con método de la secante damos dos valores iniciales t_0 y t_1, y generamos $\{t_k\}$ mediante

$$t_k = t_{k-1} - \frac{(u(b,t_{k-1}) - \beta)(t_{k-1} - t_{k-2})}{u(b,t_{k-1}) - u(b,t_{k-2})}, \quad k \geq 2.$$

Se suelen tomar

$$t_0 = (\beta - \alpha)/(b-a), \quad t_1 = t_0 + (\beta - u(b,t_0))/(b-a),$$

donde t_0 es la pendiente de la recta que une los punto (a, α) y (b, β).

Para utilizar el método de Newton, damos un valor inicial t_0 (también se suele utilizar $t_0 = (\beta-\alpha)/(b-a)$) y generamos $\{t_k\}$ mediante

$$t_k = t_{k-1} - \frac{u(b, t_{k-1}) - \beta}{\frac{\partial u}{\partial t}(b, t_{k-1})}, \quad k \geq 1,$$

pero ocupamos calcular $\frac{d}{dt}u(b, t_{k-1})$ y no tenemos una expresión de la función $t \mapsto u(b, t)$.

Para encontrar $\frac{d}{dt}u(b, t_{k-1})$, reescribamos el PVI (10) como

$$\begin{aligned} &u''(x,t) = f(x, u(x,t), u'(x,t)), \quad a \leq x \leq b, \\ &u(a,t) = \alpha, \ u'(a,t) = t, \end{aligned}$$

para visualizar mejor la dependencia del parámetro t. Derivando la ecuación diferencial con respecto a t, obtenemos

$$\begin{aligned} \frac{\partial u''}{\partial t}(x,t) &= \frac{\partial f}{\partial t}(x, u(x,t), u'(x,t)) \\ &= \frac{\partial f}{\partial x}(x, u(x,t), u'(x,t))\frac{\partial x}{\partial t} + \frac{\partial f}{\partial y}(x, u(x,t), u'(x,t))\frac{\partial u}{\partial t}(x,t) \\ &+ \frac{\partial f}{\partial z}(x, u(x,t), u'(x,t))\frac{\partial u'}{\partial t}(x,t) \\ &= \frac{\partial f}{\partial y}(x, u(x,t), u'(x,t))\frac{\partial u}{\partial t}(x,t) + \frac{\partial f}{\partial z}(x, u(x,t), u'(x,t))\frac{\partial u'}{\partial t}(x,t) \end{aligned} \tag{13}$$

con las condiciones inicales

$$\frac{\partial u}{\partial t}(a,t) = 0, \quad \frac{\partial u'}{\partial t}(a,t) = 1.$$

Haciendo $w(x,t) := \frac{\partial u}{\partial t}(x,t)$, y suponiendo que las derivadas parciales con respecto a t y x conmutan en la ecuación (13), obtenemos

$$\begin{aligned} w''(x,t) &= \frac{\partial f}{\partial y}(x,u,u')w(x,t) + \frac{\partial f}{\partial z}(x,u,u')w'(x,t), \quad a \leq x \leq b, \\ w(a,t) &= 0, \quad w'(a,t) = 1. \end{aligned} \tag{14}$$

Si resolvemos este PVI, podemos encontrar el valor de $w(b,t) := \frac{\partial u}{\partial t}(b,t)$.

De manera que en cada iteración del método de Newton estaríamos resolviendo dos PVI, y la expresión de t_k estaría dada por

$$t_k = t_{k-1} - \frac{u(b, t_{k-1}) - \beta}{w(b, t_{k-1})}, \quad k \geq 1.$$

3.1.1 Algoritmo

En los siguientes algoritmos la función `RungeKutta`, al resolver un sistema de orden m, regresa una matriz $U = (u_{ij})$ de $(n+1) \times m$ tal que u_{ij} es la j-ésima solución del sistema evaluada en el punto $x_i := a + i(b-a)/n$, $j = 0, \ldots, m-1$, $i = 0, \ldots, n$ (ver Apéndice A para ver su implementación).

El método del disparo mediante el método de la secante recibe la función f que define la ecuación diferencial, los extremos del intervalo $[a,b]$, las condiciones de frontera $u(a) = \alpha$ y $u(b) = \beta$, el número n de divisiones del intervalo $[a,b]$, el número máximo de iteraciones max_iter, y una toleracia ϵ para saber si $u(b, t_k)$ ya está suficientemente cerca de β (i.e., si $|u(b,t_k) - \beta| \leq \epsilon$). Cuando se cumpla esta condición de paro, regresa la función $u(x) := u(x, t_k)$ evaluada en los puntos $x_i := a + i(b-a)/n$, $i = 0, \ldots, n$.

function MetodoDisparoNoLineal-Secante($f(x,y,z), a, b, \alpha, \beta, n, max\_iter, \epsilon$)

// función F que define el PVI (10) en forma de sistema de dos ecuaciones diferenciales de orden 1
$F(x,u,v) = (v, f(x,u,v))$
$t_k \leftarrow (\beta - \alpha)/(b-a)$ // t_0
$conditions_k \leftarrow (\alpha, t_k)$
$U^k \leftarrow$ `RungeKutta`(F, $conditions_k, a, b, n$) // la columna 0 contiene la solución al PVI (10)
$u^k \leftarrow (U^k_{0,0}, U^k_{1,0}, \ldots, U^k_{n,0})$
if $|u^k_n - \beta| <= \epsilon$ **then** // checamos condición de paro
return u^k
end if

$t_{k+1} \leftarrow t_0 + (\beta - u^k_n)/(b-a)$ // t_1
$conditions_{k+1} \leftarrow (\alpha, t_{k+1})$
$U^{k+1} \leftarrow$ `RungeKutta`(F, $conditions_{k+1}, a, b, n$) // la columna 0 contiene la solución al PVI (10)
$u^{k+1} \leftarrow (U^{k+1}_{0,0}, U^{k+1}_{1,0}, \ldots, U^{k+1}_{n,0})$

if $|u^{k+1}_n - \beta| <= \epsilon$ **then** // checamos condición de paro
return u^{k+1}
end if

for $i \in [1, \ldots, max\_iter]$ **do**

`copy` $\leftarrow t_{k+1}$

$$t_{k+1} \leftarrow t_{k+1} - \frac{(u^{k+1}_n - \beta)(t_{k+1} - t_k)}{u^{k+1}_n - u^k_n}$$

$t_k \leftarrow$ `copy`
$u^k = u^{k+1}$
$conditions_{k+1} \leftarrow (\alpha, t_{k+1})$
$U^{k+1} \leftarrow$ `RungeKutta`(F, $conditions_{k+1}, a, b, n$) // la columna 0 contiene la solución al PVI (10)
$u^{k+1} \leftarrow (U^{k+1}_{0,0}, U^{k+1}_{1,0}, \ldots, U^{k+1}_{n,0})$

if $|u^{k+1}_n - \beta| <= \epsilon$ **then** // checamos condición de paro
return u^{k+1}
end if
end for
return El método no convergió: máximo número de iteraciones alcanzadas.
end function

El método del disparo mediante el método de Newton recibe la función f que define la ecuación diferencial, junto con $\frac{\partial f}{\partial y}$ y $\frac{\partial f}{\partial y}$, y los demás parámetros que recibe el algoritmo usando el método de la secante. Tiene la misma condición de paro y regresa la aproximación a la solución $u(x, t_k)$ evaluada en los puntos $x_i := a + i(b-a)/n$, $i = 0, \ldots, n$.

function MetodoDisparoNoLineal-Newton($f(x,y,z), \frac{\partial f}{\partial y}(x,y,z), \frac{\partial f}{\partial z}(x,y,z), a, b, \alpha, \beta, n, max\_iter, \epsilon$)

// función F que define el PVI (10) en forma de sistema de dos ecuaciones diferenciales de orden 1
$F(x,u,v) = (v, f(x,u,v))$
$t_k \leftarrow (\beta - \alpha)/(b-a)$ // t_0

for $i \in [1, \ldots, max\_iter]$ **do**

$conditions_k \leftarrow (\alpha, t_k)$
$U \leftarrow$ `RungeKutta`(F, $conditions_k, a, b, n$) // la columna 0 contiene la solución al PVI (10)
$u \leftarrow (U_{0,0}, U_{1,0}, \ldots, U_{n,0})$
$du \leftarrow (U_{0,1}, U_{1,1}, \ldots, U_{n,1})$

if $|u_n - \beta| <= \epsilon$ **then** // checamos condición de paro
return u
end if

// Resolvemos el PVI auxiliar (14) que nos sirve para la iteración de Newton
// Adaptamos Runge-Kutta para utilizar las aproximaciones de $u(x, t_k)$ y $u'(x, t_k)$ obtenidas al principio de la iteración
$h \leftarrow (b-a)/n$
$x_j \leftarrow a$
// $W_{j,0}$ y $W_{j,1}$ serán la solución al PVI auxiliar y su derivada, evaluadas en $x_j = a + jh$, resp.
$(W_{0,0}, W_{0,1}) \leftarrow (0, 1)$ // condiciones iniciales del PVI auxiliar

for $i \in [1, \ldots, n]$ **do**
$k_{1,1} \leftarrow h \cdot W_{j-1,1}$
$k_{1,2} \leftarrow h \cdot (\frac{\partial f}{\partial y}(x_j, u_{j-1}, du_{j-1}) W_{j-1,0} + \frac{\partial f}{\partial z}(x_j, u_{j-1}, du_{j-1}) W_{j-1,1})$
$k_{2,1} \leftarrow h \cdot (W_{j-1,1} + k_{1,2}/2)$
$k_{2,2} \leftarrow h \cdot [\frac{\partial f}{\partial y}(x_j + h/2, u_{j-1}, du_{j-1})(W_{j-1,0} + k_{1,1}/2) + \frac{\partial f}{\partial z}(x_j + h/2, u_{j-1}, du_{j-1})(W_{j-1,1} + k_{1,2}/2)]$
$k_{3,1} \leftarrow h \cdot (W_{j-1,1} + k_{2,2}/2)$
$k_{3,2} \leftarrow h \cdot [\frac{\partial f}{\partial y}(x_j + h/2, u_{j-1}, du_{j-1})(W_{j-1,0} + k_{2,1}/2) + \frac{\partial f}{\partial z}(x_j + h/2, u_{j-1}, du_{j-1})(W_{j-1,1} + k_{2,2}/2)]$
$k_{4,1} \leftarrow h \cdot (W_{j-1,1} + k_{3,2}/2)$
$k_{4,2} \leftarrow h \cdot [\frac{\partial f}{\partial y}(x_j + h, u_j, du_j)(W_{j-1,0} + k_{3,1}/2) + \frac{\partial f}{\partial z}(x_j + h, u_j, du_j)(W_{j-1,1} + k_{3,2}/2)]$

$W_{j,0} \leftarrow W_{j-1,0} + (k_{1,1} + 2k_{2,1} + 2k_{3,1} + k_{4,1})/6$
$W_{j,1} \leftarrow W_{j-1,1} + (k_{1,2} + 2k_{2,2} + 2k_{3,2} + k_{4,2})/6$
$x_j \leftarrow a + jh$
end for

$t_{k+1} \leftarrow t_{k+1} - \dfrac{u_n - \beta}{W_{n,0}}$ // actualización de la iteración de Newton
end for
return El método no convergió: máximo número de iteraciones alcanzadas.
end function

En el notebook de Python adjunto se muestra el código de ambos algoritmos y se resuelven ejemplos.

A Apéndice

A.1 Método de Runge-Kutta

Un sistema de orden m de problemas del valor inicial de 1er orden tiene la forma

$$\begin{aligned} u_0'(t) &= f_0(t, u_0, u_1, \ldots, u_{m-1}), \\ u_1'(t) &= f_1(t, u_0, u_1, \ldots, u_{m-1}), \\ &\vdots \\ u_{m-1}'(t) &= f_{m-1}(t, u_0, u_1, \ldots, u_{m-1}), \end{aligned} \tag{15}$$

para $a \leq t \leq b$, con condiciones iniciales

$$u_0(a) = \alpha_0, u_1(a) = \alpha_1, \ldots, u_{m-1}(a) = \alpha_{m-1}. \tag{16}$$

Se resuelve el sistema encontrando funciones $u_0, \ldots, u_{m-1}$ que satisfagan las ecuaciones diferenciales (15) junto con las condiciones iniciales (16). El método de Runge-Kutta (de 4to orden) nos permite resolver dichos sistemas.

A continuación se presenta el pseudocódigo del método de Runge-Kutta (de 4to orden). El método recibe la funcion $F = (f_0, \ldots, f_{m-1})$ que define el sistema (15), la condición inicial $F(a) = \alpha = (\alpha_0, \ldots, \alpha_{m-1})$, los extremos del intervalo $[a, b]$ en donde se resolverá el sistema, y el número de puntos n en el que se dividirá el intervalo $[a, b]$. El algoritmo regresa la solución del sistema mediante una matriz $U = (u_{ij})$ de $(n+1) \times m$ tal que u_{ij} es la aproximación a $u_j(a + i(b-a)/n)$, $j = 0, \ldots, m-1$, $i = 0, \ldots, n$.

---

function RUNGE-KUTTA(F, α, a, b, n)
  $h \leftarrow (b-a)/n$
  $t_j = a$
  $(U_{0,0}, U_{0,1}, \ldots, U_{0,m-1}) = (\alpha_1, \ldots, \alpha_{m-1})$
  for $j \in [1, \ldots, n]$ **do**
    $K1 \leftarrow h \cdot F(t_j, U_{j-1,0}, U_{j-1,1}, \ldots, U_{j-1,m-1})$
    $K2 \leftarrow h \cdot F(t_j + h/2, U_{j-1,0} + K1_0/2, U_{j-1,1} + K1_1/2, \ldots, U_{j-1,m-1} + K1_{m-1}/2)$
    $K3 \leftarrow h \cdot F(t_j + h/2, U_{j-1,0} + K2_0/2, U_{j-1,1} + K2_1/2, \ldots, U_{j-1,m-1} + K2_{m-1}/2)$
    $K4 \leftarrow h \cdot F(t_j + h, U_{j-1,0} + K3_0, U_{j-1,1} + K3_1, \ldots, U_{j-1,m-1} + K3_{m-1})$
    for $i \in [0, \ldots, m-1]$ **do**
      $U_{j,i} \leftarrow U_{j-1,i} + ((K1_i + 2 * K2_i + 2 * K3_i + K4_i)/6)$
    end for
    $t_j \leftarrow a + jh$
  end for
  return U
end function

B Referencias

- Burden, R.L. and Faires, J.D. (2011) Numerical Analysis. 9th Edition, Brookscole, Boston, 259-253.
- Keller, H. B., Numerical methods for two-point boundary-value problems, Blaisdell, Waltham, MA, 1968, 184 pp. QA372.K42