Sección 3.1

---

1. Construir el método de Taylor de orden 3 para $d = 1$ y comprobar que al aplicarlo al problema $y' = 2xy$, $y(0) = 1$, lleva a la iteración

$$y_{n+1} = y_n + h\big(2x_n y_n + y_n(1 + 2x_n^2)h + 2x_n y_n(3 + 2x_n^2)\frac{h^2}{3}\big).$$

Solución. El desarrollo de Taylor produce

$$y(x_{n+1}) = y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(x_n) + \frac{h^3}{3!}y'''(x_n) + \frac{h^4}{4!}y^{(4)}(\xi_n).$$

Para calcular $y'(x_n)$, $y''(x_n)$ e $y'''(x_n)$ usamos la ecuación diferencial,

$$\begin{aligned}
y'(x) &= f(x, y(x)),\\
y''(x) &= f_x(x, y(x)) + f_y(x, y(x))y'(x)\\
&= f_x(x, y(x)) + f_y(x, y(x))f(x, y(x)),\\
y'''(x) &= f_{xx}(x, y(x)) + 2f_{xy}(x, y(x))y'(x) + f_{yy}(x, y(x))(y'(x))^2\\
&\quad + f_y(x, y(x))y''(x)\\
&= f_{xx}(x, y(x)) + 2f_{xy}(x, y(x))f(x, y(x))\\
&\quad + f_{yy}(x, y(x))(f(x, y(x)))^2\\
&\quad + f_y(x, y(x))(f_x(x, y(x)) + f_y(x, y(x))f(x, y(x))).
\end{aligned}$$

Si despreciamos el residuo, $R_n = \frac{h^4}{4!}y^{(4)}(\xi_n)$, obtenemos el método

$$\begin{aligned}
y_{n+1} = {} & y_n + hf(x_n, y_n) + \tfrac{h^2}{2}(f_x(x_n, y_n) + f_y(x_n, y_n)f(x_n, y_n))\\
& + \tfrac{h^3}{3!}\left(f_{xx}(x_n, y_n) + 2f_{xy}(x_n, y_n)f(x_n, y_n)\right.\\
& + f_{yy}(x_n, y_n)(f(x_n, y_n))^2\\
& \left. + f_y(x_n, y_n)(f_x(x_n, y_n) + f_y(x_n, y_n)f(x_n, y_n))\right).
\end{aligned}$$

Para el problema mencionado se tiene que

$$f(x, y) = 2xy, \qquad f_x(x, y) = 2y, \qquad f_y(x, y) = 2x,$$

$$f_{xx}(x, y) = 0, \qquad f_{x,y}(x, y) = 2, \qquad f_{yy}(x, y) = 0,$$

lo que introducido en la fórmula del método produce el resultado.

---

2. Construir el método de Taylor de orden 2 para problemas de dimensión d arbitraria y elaborar un programa de Matlab que resuelva el problema (PVI) por medio de dicho método.

   Sintaxis: `y=taylororden2(ld,dxld,jyld,x,y0)`, siendo `dxld` el nombre del fichero que contiene la derivada con respecto a x de la función del lado derecho de la EDO en el punto (x, y) y `jyld` el nombre del fichero que contiene el jacobiano con respecto a y de la función del lado derecho de la EDO en el punto (x, y).

Solución. El desarrollo de Taylor produce

$$y(x_{n+1}) = y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(x_n) + \frac{h^3}{3!}y'''(\xi_n).$$

Para calcular $y'(x_n)$ e $y''(x_n)$ usamos la ecuación diferencial,

$$\begin{aligned} y'(x) &= f(x, y(x)), \\ y''(x) &= f_x(x, y(x)) + f_{y^1}(x, y(x))(y^1)'(x) + f_{y^2}(x, y(x))(y^2)'(x) \\ &= f_x(x, y(x)) + f_{y^1}(x, y(x))f^1(x, y(x)) \\ &\quad + f_{y^2}(x, y(x))f^2(x, y(x)). \end{aligned}$$

Si despreciamos el residuo, $R_n = \frac{h^3}{3!}y'''(\xi_n)$, obtenemos el método

$$\begin{aligned} y_{n+1} &= y_n + hf(x_n, y_n) + \tfrac{h^2}{2}\left(f_x(x_n, y_n) + f_{y^1}(x_n, y_n)f^1(x_n, y_n)\right. \\ &\quad \left. + f_{y^2}(x_n, y_n)f^2(x_n, y_n)\right). \end{aligned}$$

En cuanto al programa, podría ser por ejemplo este.

```
% Programa que resuelve EDO por el método de Taylor de orden 2
%
% Variables de entrada:
%
% ld: Nombre del fichero que contiene la función que calcula el lado
% derecho de la EDO. Tendrá siempre dos argumentos de entrada, x
% (escalar) e y (vector columna). La salida será un vector columna.
% dxld: Nombre del fichero que contiene la función que calcula
% la derivada con respecto a x del lado derecho de la EDO. Misma
% estructura que ld.
% jyld: Nombre del fichero que contiene el jacobiano con respecto a y del
% lado derecho de la EDO. Tendrá siempre dos argumentos de entrada, x
% (escalar) e y (vector columna). La salida será una matriz de
% dimensión d x d.
% x: Malla (vector fila de longitud N).
% y0: Dato inicial (vector columna de tamaño d, la dimensión del sistema
% de edos).
%
% Variables de salida
%
% y: Solución numérica en los puntos de la malla (matriz de tamaño
% d x N).


function y=taylororden2(ld,dxld,jyld,x,y0)

N=length(x)-1;
y(:,1)=y0;
for n=1:N,
    h=x(n+1)-x(n);
    f=feval(ld,x(n),y(:,n));
    fx=feval(dxld,x(n),y(:,n));
    fy=feval(jyld,x(n),y(:,n));
    y(:,n+1)=y(:,n)+h*f+h^2*(fx+fy*f)/2;
end
```

Sección 3.2

1. Escribir el par predictor-corrector Euler/Trapecio, (3.3)–(3.4), como un método de Runge-Kutta. Visto de esta forma se le conoce como método de Euler mejorado.

Solución. El método tiene dos evaluaciones de función,

$$k_1 = f(x_n, y_n), \qquad k_2 = f(x_n + h, y_n + hk_1),$$

a partir de las cuales se calcula y_{n+1} por medio de la fórmula

$$y_{n+1} = y_n + \frac{h}{2}(k_1 + k2).$$

Así pues, estamos ante un método de Runge-Kutta de tablero

$$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & 1/2 & 1/2. \end{array}$$

2. Consideramos el problema (PVI) con una f tal que $c^T f(x, y) = 0$ para un cierto vector columna c. Según vimos en un problema del capítulo anterior, en estas condiciones la solución del problema satisface la ley de conservación lineal $c^T y(x) = c^T \eta$ para todo $x \in [a, b]$. Estudiar si las soluciones obtenidas al aplicar al problema un método de Runge-Kutta satisfacen la misma ley de conservación lineal, $c^T y_n = c^T y_0$, $n = 0, \ldots, N$.

Solución. Basta con observar que

$$c^T k_i = c^T f(x_n + c_i h, y_n + h \sum_{j=1}^{s} a_{ij} k_j) = 0, \qquad i = 1, \ldots, s.$$

Por consiguiente,

$$c^T y_{n+1} = c^T y_n + h \sum_{i=1}^{s} b_i c^T k_i = c^T y_n,$$

de donde se deduce inmediatamente el resultado.

---

3. Si integramos la EDO en (x_n, x_{n+1}) y aplicamos la regla de Simpson, obtenemos que

$$y(x_{n+1}) - y(x_n) \approx \frac{h}{6}(f(x_n, y(x_n)) + 4f(x_{n+\frac{1}{2}}, y(x_{n+\frac{1}{2}})) + f(x_{n+1}, y(x_{n+1}))),$$

donde $x_{n+\frac{1}{2}} = x_n + \frac{h}{2}$. Como no conocemos el valor de $y(x_{n+\frac{1}{2}})$, aproximamos esta cantidad por Euler, $y(x_{n+\frac{1}{2}}) \approx y(x_n) + \frac{h}{2} f(x_n, y(x_n))$. Tampoco conocemos el valor de $y(x_{n+1})$. Para estimarlo podemos avanzar a partir de x_n utilizando una media ponderada de las dos pendientes calculadas hasta ahora, $k_1 = f(x_n, y(x_n))$, $k_2 = f(x_{n+\frac{1}{2}}, y(x_n) + \frac{h}{2}k_1)$; es decir, tomamos $y(x_{n+1}) \approx y(x_n) + h(\alpha k_1 + \beta k_2)$. Llegamos así al método

$$\begin{cases} k_1 = f(x_n, y_n), \\ k_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1), \\ k_3 = f(x_n + h, y_n + h(\alpha k_1 + \beta k_2)) \\ y_{n+1} = y_n + \frac{h}{6}(k_1 + 4k_2 + k_3). \end{cases}$$

Determinar α y β para que $\|R_n\| \leq Ch^4$ para alguna constante C.

Solución. El residuo viene dado por

$$R_n = F(h) - G(h)$$

con

$$F(h) = y(x_n + h)$$

$$G(h) = y(x_n) + \tfrac{h}{6}\Big(k_1(x_n, y(x_n); h) + 4k_2(x_n, y(x_n); h) + k_3(x_n, y(x_n); h)\Big).$$

Si desarrollamos $F(h)$ en potencias de h, se tiene que

$$F(h) = y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(x_n) + \frac{h^3}{3!}y'''(x_n) + O(h^4).$$

Usando la EDO,

$$\begin{aligned} F(h) &= y(x_n) + hf(x_n, y(x_n)) + \tfrac{h^2}{2} f_x(x_n, y(x_n)) \\ &\quad + \tfrac{h^2}{2} \sum_{J=1}^{d} f_{y^J}(x_n, y(x_n)) f^J(x_n, y(x_n)) \\ &\quad + \tfrac{h^3}{3!} f_{xx}(x_n, y(x_n)) + \tfrac{h^3}{3} \sum_{J=1}^{d} f_{xy^J}(x_n, y(x_n)) f^J(x_n, y(x_n)) \\ &\quad + \tfrac{h^3}{3!} \sum_{J,L=1}^{d} f_{y^J y^L}(x_n, y(x_n)) f^J(x_n, y(x_n)) f^L(x_n, y(x_n)) \\ &\quad + \tfrac{h^3}{3!} \sum_{J=1}^{d} f_{y^J}(x_n, y(x_n)) f_x^J(x_n, y(x_n)) \\ &\quad + \tfrac{h^3}{3!} \sum_{J,L=1}^{d} f_{y^J}(x_n, y(x_n)) f_{y^L}^J(x_n, y(x_n)) f^L(x_n, y(x_n)) \\ &\quad + O(h^4) \end{aligned}$$

Para obtener el desarrollo de $G(h)$ desarrollamos las etapas. El desarrollo de la primera es trivial,

$$k_1(x_n, y(x_n); h) = f(x_n, y(x_n)).$$

En cuanto a la segunda, tenemos que

$$\begin{aligned} & k_2(x_n, y(x_n); h) = f(x_n + \tfrac{h}{2}, y(x_n) + \tfrac{h}{2} f(x_n, y(x_n))), \\ & \tfrac{dk_2}{dh}(x_n, y(x_n); h) = \tfrac{1}{2} f_x(x_n + \tfrac{h}{2}, y(x_n) + \tfrac{h}{2} f(x_n, y(x_n)))) \\ & + \tfrac{1}{2} \sum_{J=1}^{d} f_{y^J}(x_n + \tfrac{h}{2}, y(x_n) + \tfrac{h}{2} f(x_n, y(x_n))) f^J(x_n, y(x_n)), \\ & \tfrac{d^2k_2}{dh^2}(x_n, y(x_n); h) = \tfrac{1}{4} f_{xx}(x_n + \tfrac{h}{2}, y(x_n) + \tfrac{h}{2} f(x_n, y(x_n)))) \\ & + \tfrac{2}{4} \sum_{J=1}^{d} f_{xy^J}(x_n + \tfrac{h}{2}, y(x_n) + \tfrac{h}{2} f(x_n, y(x_n))) f^J(x_n, y(x_n)) \\ & + \tfrac{1}{4} \sum_{J,L=1}^{d} f_{y^J y^L}(x_n + \tfrac{h}{2}, y(x_n) + \tfrac{h}{2} f^J(x_n, y(x_n))) f^J(x_n, y(x_n)) f^L(x_n, y(x_n)). \end{aligned}$$

Por consiguiente,

$$k_2(x_n, y(x_n); 0) = f(x_n, y(x_n)),$$

$$\frac{dk_2}{dh}(x_n, y(x_n); 0) = \frac{1}{2} f_x(x_n, y(x_n)) + \frac{1}{2} \sum_{J=1}^{d} f_{y^J}(x_n, y(x_n)) f^J(x_n, y(x_n)),$$

$$\frac{d^2k_2}{dh^2}(x_n, y(x_n); 0) = \frac{1}{4} f_{xx}(x_n, y(x_n)) + \frac{1}{2} \sum_{J=1}^{d} f_{xy^J}(x_n, y(x_n)) f^J(x_n, y(x_n))$$
$$+ \frac{1}{4} \sum_{J,L=1}^{d} f_{y^J y^L}(x_n, y(x_n)) f^J(x_n, y(x_n)) f^L(x_n, y(x_n)).$$

A partir de esto es fácil obtener el desarrollo requerido de k_2.

En cuanto a la tercera etapa, la escribimos en la forma

$$k_3(x_n, y(x_n); h) = f(x_n + h, a(h)),$$

donde

$$a(h) = y(x_n) + h(\alpha f(x_n, y(x_n)) + \beta k_2(x_n, y(x_n); h)).$$

Se tiene que

$$\frac{dk_3}{dh}(x_n, y(x_n); h) = f_x(x_n + h, a(h)) + \sum_{J=1}^{d} f_{y^J}(x_n + h, a(h))(a^J)'(h)$$

$$\frac{d^2k_3}{dh^2}(x_n, y(x_n); h) = f_{xx}(x_n + h, a(h)) + 2 \sum_{J=1}^{d} f_{xy^J}(x_n + h, a(h))(a^J)'(h)$$
$$+ \sum_{J,L=1}^{d} f_{y^J y^L}(x_n + h, a(h))(a^J)'(h)(a^L)'(h)$$
$$+ \sum_{J=1}^{d} f_{y^J}(x_n + h, a(h))(a^J)''(h).$$

Nótese que

$$a'(h) = \alpha f(x_n, y(x_n)) + \beta k_2(x_n, y(x_n); h) + h\beta \frac{dk_2}{dh}(x_n, y(x_n); h),$$

$$a''(h) = 2\beta \frac{dk_2}{dh}(x_n, y(x_n); h) + h\beta \frac{d^2k_2}{dh^2}(x_n, y(x_n); h),$$

y por tanto que

$$a'(0) = \alpha f(x_n, y(x_n)) + \beta k_2(x_n, y(x_n); 0) = (\alpha + \beta) f(x_n, y(x_n)),$$

$$a''(0) = \beta f_x(x_n, y(x_n)) + \beta \sum_{L=1}^{d} f_{y^L}(x_n, y(x_n)) f^L(x_n, y(x_n)).$$

Llegamos así a

$$k_3(x_n, y(x_n); 0) = f(x_n, y(x_n))$$

$$\frac{dk_3}{dh}(x_n, y(x_n); 0) = f_x(x_n, y(x_n)) + (\alpha+\beta)\sum_{J=1}^{d} f_{y^J}(x_n, y(x_n)) f^J(x_n, y(x_n))$$

$$\frac{d^2k_3}{dh^2}(x_n, y(x_n); 0) = f_{xx}(x_n, y(x_n)) + 2(\alpha+\beta)\sum_{J=1}^{d} f_{xy^J}(x_n, y(x_n)) f^J(x_n, y(x_n))$$
$$+(\alpha+\beta)^2 \sum_{J,L=1}^{d} f_{y^J y^L}(x_n, y(x_n)) f^J(x_n, y(x_n)) f^L(x_n, y(x_n))$$
$$+\beta \sum_{J=1}^{d} f_{y^J}(x_n, y(x_n)) f_x^J(x_n, y(x_n))$$
$$+\beta \sum_{J,L=1}^{d} f_{y^J}(x_n, y(x_n)) f_{y^L}^J(x_n, y(x_n)) f^L(x_n, y(x_n)),$$

a partir de lo cual es inmediato obtener el desarrollo de k_3.

Juntando todo lo anterior, obtenemos el siguiente desarrollo para $G(h)$,

$$\begin{aligned} G(h) \;=\; & y(x_n) + hf(x_n, y(x_n)) + \tfrac{h^2}{2} f_x(x_n, y(x_n)) \\ & + \tfrac{(2+\alpha+\beta)h^2}{6} \sum_{J=1}^{d} f_{y^J}(x_n, y(x_n)) f^J(x_n, y(x_n)) \\ & + \tfrac{h^3}{3!} f_{xx}(x_n, y(x_n)) + \tfrac{(1+\alpha+\beta)h^3}{6} \sum_{J=1}^{d} f_{xy^J}(x_n, y(x_n)) f^J(x_n, y(x_n)) \\ & + \tfrac{(1+(\alpha+\beta)^2)h^3}{12} \sum_{J,L=1}^{d} f_{y^J y^L}(x_n, y(x_n)) f^J(x_n, y(x_n)) f^L(x_n, y(x_n)) \\ & + \tfrac{\beta h^3}{2} \sum_{J=1}^{d} f_{y^J}(x_n, y(x_n)) f_x^J(x_n, y(x_n)) \\ & + \tfrac{\beta h^3}{2} \sum_{J,L=1}^{d} f_{y^J}(x_n, y(x_n)) f_{y^L}^J(x_n, y(x_n)) f^L(x_n, y(x_n)) \\ & + O(h^4) \end{aligned}$$

Por consiguiente, los desarrollos de F y G coinciden hasta orden 3, y por tanto $R_n = O(h^4)$, si y sólo si $\alpha + \beta = 1$ y $\beta = 1/3$, es decir, si

$$\alpha = \frac{2}{3}, \qquad \beta = \frac{1}{3}.$$

4. Consideramos la igualdad

$$y(x_{n+k}) - y(x_{n+k-1}) = \int_{x_{n+k-1}}^{x_{n+k}} f(t, y(t))\, dt. \tag{1}$$

Si aproximamos la integral sustituyendo el integrando por el polinomio $p_{n,k}(t)$ de grado menor o igual que $k-1$ que interpola a $f(t, y(t))$ en los nodos $x_n, \ldots, x_{n+k-1}$, y pedimos que la aproximación y_n satisfaga la relación resultante exactamente, se obtiene el método de Adams-Bashforth de k pasos.

(a) Demostrar que el método de Adams-Bashforth de k pasos para mallas equiespaciadas viene dado por la recurrencia

$$y_{n+k} - y_{n+k-1} = h \sum_{j=0}^{k-1} \beta_{kj} f(x_{n+j}, y_{n+j}),$$

donde los coeficiente β_{kj} satisfacen

$$\beta_{kj} = \int_{k-1}^{k} \prod_{l=0,\ l\neq j}^{k-1} \frac{s-l}{j-l}\, ds.$$

(b) Usando la fórmula para el error de interpolación, obtener una expresión para el residuo del método,

$$\begin{aligned} R_n &= y(x_{n+k}) - y(x_{n+k-1}) - \int_{x_{n+k-1}}^{x_{n+k}} p_{n,k}(t)\, dt \\ &= \int_{x_{n+k-1}}^{x_{n+k}} (y'(t) - p_{n,k}(t))\, dt, \end{aligned}$$

que demuestre que $\|R_n\| \leq Ch^{k+1}$.

(c) Obtener el método de Adams-Bashforth de 2 pasos para mallas generales, no necesariamente equiespaciadas, y una expresión para su residuo.

Solución. (a) Estamos haciendo la aproximación

$$f(t, y(t)) \approx \sum_{j=0}^{k-1} f(x_{n+j}, y(x_{n+j}))L_j(t), \qquad L_j(t) = \prod_{l=0,\ l\neq j}^{k-1} \frac{t - x_{n+l}}{x_{n+j} - x_{n+l}}.$$

La integral de los polinomios de Lagrange $L_j(t)$ se calcula haciendo el cambio de variables $t = x_n + sh$; llegamos así a

$$y(x_{n+k}) - y(x_{n+k-1}) \approx h \sum_{j=0}^{k-1} \beta_{kj} f(x_{n+j}, y(x_{n+j}))\, dt, \tag{2}$$

donde

$$\beta_{kj} = \int_{k-1}^{k} \prod_{l=0,\ l\neq j}^{k-1} \frac{s-l}{j-l}\, ds.$$

El método de Adams explícito de k pasos se obtiene pidiendo que las aproximaciones $y_j \approx y(x_j)$ satisfagan la relación (2) exactamente, es decir,

$$y_{n+k} - y_{n+k-1} = h \sum_{j=0}^{k-1} \beta_{kj} f_{n+j}.$$

(b) Usando la fórmula del error de interpolación obtenemos

$$y'(t) - p_{n,k}(t) = \frac{y^{(k+1)}(\overline{\xi})}{k!} \prod_{l=0}^{k-1} (t - x_{n+l}),$$

donde $\overline{\xi} \in (x_n, t)$ puede variar de componente a componente.

Por otra parte, $\prod_{l=0}^{k-1} (t - x_{n+l})$ no cambia de signo en el intervalo de integración (x_{n+k-1}, x_{n+k}); podemos aplicar el Teorema del Valor Medio para integrales, y llegamos a

$$\begin{aligned} R_n &= \int_{x_{n+k-1}}^{x_{n+k}} \frac{y^{(k+1)}(\overline{\xi}(t))}{k!} \prod_{l=0}^{k-1} (t - x_{n+l})\, dt \\ &= \frac{y^{(k+1)}(\overline{\theta})}{k!} \int_{x_{n+k-1}}^{x_{n+k}} \prod_{l=0}^{k-1} (t - x_{n+l})\, dt. \end{aligned}$$

Haciendo el cambio de variables $t = x_n + sh$ concluimos que

$$R_n = \frac{y^{(k+1)}(\overline{\theta})}{k!}\, h^{k+1} \int_{k-1}^{k} \prod_{l=0}^{k-1} (s-l)\, ds = C(k) y^{(k+1)}(\overline{\theta}) h^{k+1},$$

y el método es consistente de orden $p = k$.

Los cuatro primeros métodos de Adams-Bashforth son

$$\begin{aligned}
&y_{n+1} - y_n = hf_n,\\
&y_{n+2} - y_{n+1} = \tfrac{h}{2}(3f_{n+1} - f_n),\\
&y_{n+3} - y_{n+2} = \tfrac{h}{12}(23f_{n+2} - 16f_{n+1} + 5f_n),\\
&y_{n+4} - y_{n+3} = \tfrac{h}{24}(55f_{n+3} - 59f_{n+2} + 37f_{n+1} - 9f_n).
\end{aligned}$$

Como vemos, con $k = 1$ se tiene el método de Euler.

(c) En este caso hacemos la aproximación $f(t, y(t)) \approx p(t)$, donde

$$p(t) = f(x_n, y(x_n))\frac{t - x_{n+1}}{x_n - x_{n+1}} + f(x_{n+1}, y(x_{n+1}))\frac{t - x_n}{x_{n+1} - x_n}.$$

Así,

$$\begin{aligned}
&y(x_{n+2}) - y(x_{n+1}) \approx \textstyle\int_{x_{n+1}}^{x_{n+2}} p(t)\,dt\\
&= -\tfrac{(x_{n+2}-x_{n+1})^2}{2(x_{n+1}-x_n)} f(x_n, y(x_n)) + \tfrac{(x_{n+2}-x_n)^2-(x_{n+1}-x_n)^2}{2(x_{n+1}-x_n)} f(x_{n+1}, y(x_{n+1})).
\end{aligned}$$

El método pedido es por tanto

$$y_{n+2} - y_{n+1} = -\tfrac{(x_{n+2}-x_{n+1})^2}{2(x_{n+1}-x_n)} f_n + \tfrac{(x_{n+2}-x_n)^2-(x_{n+1}-x_n)^2}{2(x_{n+1}-x_n)} f_{n+1}.$$

Usando la fórmula del error de interpolación obtenemos

$$y'(t) - p(t) = \frac{y^{(3)}(\overline{\xi})}{2}\,(t - x_n)(t - x_{n+1}),$$

donde $\overline{\xi} \in (x_n, t)$ puede variar de componente a componente. Como $(t - x_n)(t - x_{n+1}) > 0$ en el intervalo de integración (x_{n+1}, x_{n+2}); podemos aplicar el Teorema del Valor Medio para integrales, y llegamos a

$$\begin{aligned}
R_n &= \int_{x_{n+1}}^{x_{n+2}} \frac{y^{(3)}(\overline{\xi}(t))}{2}\,(t - x_n)(t - x_{n+1})\,dt\\
&= \frac{y^{(3)}(\overline{\theta})}{2}\int_{x_{n+1}}^{x_{n+2}} (t - x_n)(t - x_{n+1})\,dt\\
&= \frac{y^{(3)}(\overline{\theta})}{2}(x_{n+2} - x_{n+1})^2\left(\frac{x_{n+2} - x_n}{2} - \frac{x_{n+2} - x_{n+1}}{3!}\right).
\end{aligned}$$

Comprobamos así que $R_n = O(h^3)$, con $h = \max_n h_n$.

5. Si aproximamos la integral de la igualdad (1) sustituyendo el integrando por el polinomio $\hat{p}_{n,k}(t)$ de grado menor o igual que k que interpola a $f(t, y(t))$ en los nodos $x_n, \ldots, x_{n+k}$, y pedimos que la aproximación y_n satisfaga la relación resultante exactamente, se obtiene el método de Adams-Moulton de k pasos.

   (a) Demostrar que el método de Adams-Moulton de k pasos para mallas equiespaciadas viene dado por la recurrencia

   $$y_{n+k} - y_{n+k-1} = h \sum_{j=0}^{k} \beta^*_{kj} f(x_{n+j}, y_{n+j}), \tag{3}$$

   donde los coeficiente β_{kj} satisfacen

   $$\beta^*_{kj} = \int_{k-1}^{k} \prod_{l=0,\ l\neq j}^{k} \frac{s-l}{j-l}\, ds.$$

   (b) Usando la fórmula para el error de interpolación, obtener una expresión para el residuo del método,

   $$R_n = \int_{x_{n+k-1}}^{x_{n+k}} (y'(t) - \hat{p}_{n,k}(t))\, dt,$$

   que demuestre que $\|R_n\| \leq Ch^{k+2}$.

   (c) Dar una condición sobre h que garantice que el sistema (3) tiene una única solución y_{n+k}.

Solución. (a) Estamos haciendo la aproximación

$$f(t, y(t)) \approx \sum_{j=0}^{k} f(x_{n+j}, y(x_{n+j}))L_j(t), \qquad L_j(t) = \prod_{l=0,\ l\neq j}^{k} \frac{t - x_{n+l}}{x_{n+j} - x_{n+l}}.$$

La integral de los polinomios de Lagrange $L_j(t)$ se calcula haciendo el cambio de variables $t = x_n + sh$; llegamos así a

$$y(x_{n+k}) - y(x_{n+k-1}) \approx h \sum_{j=0}^{k} \beta^*_{kj} f(x_{n+j}, y(x_{n+j})),$$

con

$$\beta^*_{kj} = \int_{k-1}^{k} \prod_{l=0,\ l\neq j}^{k} \frac{s-l}{j-l}\, ds.$$

El método de Adams implícito de k pasos es entonces

$$y_{n+k} - y_{n+k-1} = h \sum_{j=0}^{k} \beta^*_{kj} f_{n+j}.$$

(b) Razonando como en el caso de los métodos explícitos obtenemos que el error de truncación viene dado por

$$\begin{aligned} R_n &= \int_{x_{n+k-1}}^{x_{n+k}} (y'(t) - \hat{p}_{n,k}(t))\, dt = \frac{y^{(k+2)}(\overline{\theta})}{(k+1)!} \int_{x_{n+k-1}}^{x_{n+k}} \prod_{l=0}^{k} (t - x_{n+l})\, dt \\ &= C(k) y^{(k+2)}(\overline{\theta}) h^{k+2}. \end{aligned}$$

(c) El valor de y_{n+k} es un punto fijo de la aplicación

$$F(y) = y_{n+k-1} + h\beta^*_{kk} f(x_{n+k}, y) + h \sum_{j=0}^{k-1} \beta^*_{kj} f(x_{n+j}, y_{n+j}).$$

Queremos ver que F es contractiva, y que por tanto tiene un único punto fijo, si h es suficientemente pequeño. En efecto,

$$\|F(y) - F(\hat{y})\| = h\beta^*_{kk} \|f(x_{n+k}, y) - f(x_{n+k}, \hat{y})\| \leq Lh\beta^*_{kk} \|y - \hat{y}\|,$$

y por tanto F es contractiva si $Lh\beta^*_{kk} < 1$.

---

6. Consideramos la igualdad

$$y(x_{n+k}) - y(x_{n+k-2}) = \int_{x_{n+k-2}}^{x_{n+k}} f(t, y(t))\, dt.$$

Si aproximamos la integral sustituyendo el integrando por el polinomio de grado menor o igual que $k-1$ que interpola a $f(t, y(t))$ en los nodos $x_n, \ldots, x_{n+k-1}$, y pedimos que la aproximación y_n satisfaga la relación resultante exactamente, se obtiene el método de Nyström[1] de k pasos.

[1] Evert Johannes Nyström (1895–1960), matemático finés conocido por sus métodos numéricos para ecuaciones integrales y para problemas de valor inicial para ecuaciones de segundo orden.

(a) Obtener una fórmula para el método de Nyström de k pasos.

(b) Usar la fórmula para el error de interpolación para demostrar que existe una constante C tal que el residuo del método de Nyström de k pasos satisface $\|R_n\| \leq Ch^{k+1}$.

Solución. (a) Estamos haciendo la aproximación

$$f(t, y(t)) \approx \sum_{j=0}^{k-1} f(x_{n+j}, y(x_{n+j}))L_j(t), \qquad L_j(t) = \prod_{l=0,\ l\neq j}^{k-1} \frac{t - x_{n+l}}{x_{n+j} - x_{n+l}}.$$

La integral de los polinomios de Lagrange $L_j(t)$ se calcula haciendo el cambio de variables $t = x_n + sh$; llegamos así a

$$y(x_{n+k}) - y(x_{n+k-2}) \approx h \sum_{j=0}^{k-1} \gamma_{kj} f(x_{n+j}, y(x_{n+j}))\, dt,$$

donde

$$\gamma_{kj} = \int_{k-2}^{k} \prod_{l=0,\ l\neq j}^{k-1} \frac{s-l}{j-l}\, ds.$$

El método de Nyström de k pasos es entonces

$$y_{n+k} - y_{n+k-2} = h \sum_{j=0}^{k-1} \gamma_{kj} f_{n+j}.$$

(b) Usando la fórmula del error de interpolación obtenemos

$$y'(t) - p_{n,k}(t) = \frac{y^{(k+1)}(\overline{\xi})}{k!} \prod_{l=0}^{k-1} (t - x_{n+l}),$$

donde $\overline{\xi} \in (x_n, t)$ puede variar de componente a componente.

Desgraciadamente, $\prod_{l=0}^{k-1}(t - x_{n+l})$ cambia de signo en el intervalo de integración (x_{n+k-1}, x_{n+k}). Así que no podemos utilizar el Teorema del

Valor Medio para Integrales. En cualquier caso, una estimación cruda produce

$$\|R_n\| \leq 2h^{k+1} \max_{x\in[a,b]} \|y'''(x)\|.$$

---

7. Programar funciones de Matlab que calculen aproximaciones numéricas para el problema (PVI) por medio de los siguientes métodos:

   (a) El método de Euler mejorado.

   (b) El método de Euler modificado.

   (c) El método de Adams-Bashforth de 2 pasos (para mallas uniformes). Hay que programar el método de manera que en cada paso (salvo el primero) sólo se haga una evaluación de función. El segundo valor de arranque, y_1, se calculará mediante el método de Euler.

   La estructura de las variables de entrada y salida será la misma que la del programa `euler` presentado en el capítulo 2.

Solución. (a)

```
function y=eulermejorado(ld,x,y0)

N=length(x)-1;
y(:,1)=y0;
for n=1:N,
    hn=(x(n+1)-x(n));
    k1=feval(ld,x(n),y(:,n));
    k2=feval(ld,x(n+1),y(:,n)+hn*k1);
    y(:,n+1)=y(:,n)+hn*(k1+k2)/2;
end
```

(b)

```
function y=eulermodificado(ld,x,y0)

N=length(x)-1;
y(:,1)=y0;
for n=1:N,
    hn=(x(n+1)-x(n));
    k1=feval(ld,x(n),y(:,n));
    k2=feval(ld,x(n)+hn/2,y(:,n)+hn*k1/2);
    y(:,n+1)=y(:,n)+hn*k2;
end
```

(c)

```
function y=ab2(ld,x,y0)

y(:,1)=y0;
N=length(x)-1;
h=x(2)-x(1);

 %Calculo de y2 por el metodo de Euler
fn=feval(ld,x(1),y(:,1));
y(:,2)=y(:,1)+h*fn;

for n=1:N-1
    fn1=feval(ld,x(n+1),y(:,n+1));
    y(:,n+2)=y(:,n+1)+h/2*(3*fn1-fn);
    fn=fn1;
end
```

Sección 3.3

1. Obtener, por medio del desarrollo de Taylor, una aproximación de orden 2 de la derivada $y'(x)$ mediante una combinación lineal de $y(x-2h)$, $y(x-h)$ e $y(x)$ (fórmula de diferenciación regresiva de tres puntos). Utilizarla para construir un método de dos pasos para resolver el PVI. Obtener una expresión para el residuo. El método construido se conoce como fórmula BDF ("backward differentiation formulae") de dos pasos.

Solución. Queremos que la combinación lineal

$$\frac{\alpha y(x) + \beta y(x-h) + \gamma y(x-2h)}{h}$$

sea igual a $y'(x) + O(h^2)$. Usando los desarrollos de Taylor

$$y(x-h) = \sum_{k=0}^{2} y^{(k)}(x)\frac{(-h)^k}{k!} - \frac{h^3}{3!}y^{(3)}(\xi), \qquad \xi \in (x-h, x),$$

$$y(x-2h) = \sum_{k=0}^{2} y^{(k)}(x)\frac{(-2h)^k}{k!} - \frac{(2h)^3}{3!}y^{(3)}(\eta), \qquad \eta \in (x-2h, x),$$

vemos que lo que se necesita es

$$\alpha + \beta + \gamma = 0, \quad -\beta - 2\gamma = 1, \quad \frac{\beta}{2} + 2\gamma = 0.$$

La solución de este sistema lineal resulta ser

$$\alpha = \frac{3}{2}, \qquad \beta = -2, \qquad \gamma = \frac{1}{2}.$$

Llegamos por tanto a la fórmula

$$y'(x) = \frac{3y(x) - 4y(x-h) + y(x-2h)}{2h} - \frac{h^2}{3}y'''(\xi) + \frac{2h^2}{3}y'''(\eta).$$

Tomando ahora $x = x_n + 2h$, obtenemos

$$\begin{aligned} f(x_{n+2}, y(x_{n+2})) = y'(x_{n+2}) &= \frac{3y(x_{n+2}) - 4y(x_{n+1}) + y(x_n)}{2h} \\ &\quad -\frac{h^2}{3}y'''(\overline{\xi}_n) + \frac{2h^2}{3}y'''(\overline{\eta}_n), \end{aligned}$$

con $\overline{\xi}_n \in (x_n, x_{n+2})$, $\overline{\eta}_n \in (x_{n+1}, x_{n+2})$. Despreciando el término de orden 2 y pidiendo que la relación se satisfaga exactamente llegamos al método

$$y_{n+2} - \frac{4}{3}y_{n+1} + \frac{1}{3}y_n = \frac{2h}{3}f(x_{n+2}, y_{n+2}),$$

siendo el residuo

$$R_n = \frac{2h^3}{9}y'''(\overline{\xi}_n) - \frac{4h^3}{9}y'''(\overline{\eta}_n).$$

---

2. Obtener una aproximación de orden 4 de la derivada $y'(x)$ a partir de los valores $y(x+2h)$, $y(x+h)$, $y(x)$, $y(x-h)$ e $y(x-2h)$ (fórmula de diferenciación centrada de cinco puntos). Utilizarla para construir un método de cuatro pasos para resolver problemas de valor inicial para EDOs. Obtener una expresión para el residuo.

Solución. Queremos que la combinación lineal

$$\frac{ay(x+2h) + by(x+h) + cy(x) + dy(x-h) + ey(x-2h)}{h}$$

sea igual a $y'(x) + O(h^4)$. Usando los desarrollos de Taylor

$$y(x+2h) = \sum_{k=0}^{4} y^{(k)}(x)\tfrac{(2h)^k}{k!} + \tfrac{(2h)^5}{5!}y^{(5)}(\theta_1), \qquad \theta_1 \in (x, x+2h),$$

$$y(x+h) = \sum_{k=0}^{4} y^{(k)}(x)\tfrac{h^k}{k!} + \tfrac{h^5}{5!}y^{(5)}(\theta_2), \qquad \theta_2 \in (x, x+h),$$

$$y(x-h) = \sum_{k=0}^{4} y^{(k)}(x)\tfrac{(-h)^k}{k!} - \tfrac{h^5}{5!}y^{(5)}(\theta_3), \qquad \theta_3 \in (x-h, x),$$

$$y(x-2h) = \sum_{k=0}^{4} y^{(k)}(x)\tfrac{(-2h)^k}{k!} - \tfrac{(2h)^5}{5!}y^{(5)}(\theta_4), \qquad \theta_4 \in (x-2h, x),$$

vemos que lo que se necesita es

$$\begin{aligned}
&a + b + c + d + e = 0,\\
&2a + b - d - 2e = 1,\\
&4a + b + d + 4e = 0,\\
&8a + b - d - 8e = 0,\\
&16a + b + d + 16e = 0.
\end{aligned}$$

La solución de este sistema lineal resulta ser

$$a = -\frac{1}{12}, \quad b = \frac{2}{3}, \quad c = 0 \quad d = -\frac{2}{3}, \quad e = \frac{1}{12}.$$

Así, llegamos a la fórmula

$$y'(x) = \frac{-y(x+2h) + 8y(x+h) - 8y(x-h) + y(x-2h)}{12h}$$

$$-\frac{h^4}{45}y^{(5)}(\theta_1) + \frac{h^4}{180}y^{(5)}(\theta_2) + \frac{h^4}{180}y^{(5)}(\theta_3) - \frac{h^4}{45}y^{(5)}(\theta_4).$$

Tomando $x = x_n + 2h$ obtenemos

$$f(x_{n+2}, y(x_{n+2})) = y'(x_{n+2}) = \frac{-y(x_{n+4}) + 8y(x_{n+3}) - 8y(x_{n+1}) + y(x_n)}{12h}$$

$$-\frac{h^4}{45}y^{(5)}(\overline{\theta}_{1n}) + \frac{h^4}{180}y^{(5)}(\overline{\theta}_{2n}) + \frac{h^4}{180}y^{(5)}(\overline{\theta}_{3n}) - \frac{h^4}{45}y^{(5)}(\overline{\theta}_{4n}),$$

con $\overline{\theta}_{1n} \in (x_{n+2}, x_{n+4})$, $\overline{\theta}_{2n} \in (x_{n+2}, x_{n+3})$, $\overline{\theta}_{3n} \in (x_{n+1}, x_{n+2})$ y $\overline{\theta}_{2n} \in (x_n, x_{n+2})$. Llegamos así al método

$$y_{n+4} - 8y_{n+3} + 8y_{n+1} - y_n = -12hf(x_{n+2}, y_{n+2}),$$

con un residuo dado por

$$R_n = -\frac{4h^5}{15}y^{(5)}(\overline{\theta}_{1n}) + \frac{h^5}{15}y^{(5)}(\overline{\theta}_{2n}) + \frac{h^5}{15}y^{(5)}(\overline{\theta}_{3n}) - \frac{4h^5}{15}y^{(5)}(\overline{\theta}_{4n}).$$

---

3. Consideramos el polinomio $p_1(x)$ de grado menor o igual que uno que pasa por los puntos $(x_n, y(x_n))$, $(x_{n+1}, y(x_{n+1}))$. Se sabe que

$$y'(x) = p_1'(x) + O(h), \qquad x \in [x_n, x_{n+1}].$$

Utilizar esta idea para construir un método numérico de un paso para resolver el PVI.

Solución. Puesto que p_1 es un polinomio de grado menor o igual que 1, su derivada es constante,

$$p_1'(x) = \frac{y(x_{n+1}) - y(x_n)}{x_{n+1} - x_n}.$$

Se tiene entonces

$$\frac{y(x_{n+1}) - y(x_n)}{x_{n+1} - x_n} = f(x, y(x)) + O(h)$$

si $x \in [x_n, x_{n+1}]$. Tomando $x = x_n$ llegamos al método de Euler, y tomando $x = x_{n+1}$ al método de Euler implícito. En ambos casos obtenemos que el residuo es una $O(h^2)$.

Sección 3.4

---

1. Dados ν *parámetros de colocación* $c_1, \ldots, c_\nu$ (preferiblemente en $[0,1]$, aunque no es imprescindible), buscamos un polinomio u de grado ν (con coeficientes vectoriales) tal que

$$u(x_n) = y_n, \qquad u'(x_n + c_j h) = f(x_n + c_j h, u(x_n + c_j h)),\ j = 1, \ldots, \nu.$$

Al calcular u y tomar $y_{n+1} = u(x_{n+1})$ se obtiene un método de colocación que pertenece a la familia de los métodos de Runge-Kutta.

Obtener el método de Runge-Kutta de colocación que corresponde a los parámetros de colocación $c_1 = 1/3$, $c_2 = 1$ (este método se conoce como método de Radau IIA de orden 3).

Solución. El polinomio u tiene grado menor o igual que 2. Por lo tanto, u' es un polinomio de grado menor o igual que 1, que pasa por los puntos

$$(x_n + \frac{h}{3}, f(x_n + \frac{h}{3}, u(x_n + \frac{h}{3}))), \qquad (x_n + h, f(x_n + h, u(x_n + h))).$$

Si definimos $k_1 = f(x_n + \frac{h}{3}, u(x_n + \frac{h}{3}))$, $k_2 = f(x_n + h, u(x_n + h))$, es fácil ver entonces que

$$\begin{aligned} u'(x) &= \frac{x - (x_n + h)}{(x_n + \frac{h}{3} - (x_n + h))} k_1 + \frac{x - (x_n + \frac{h}{3})}{(x_n + h - (x_n + \frac{h}{3}))} k_2 \\ &= -\frac{3}{2h}(x - (x_n + h))k_1 + \frac{3}{2h}(x - (x_n + \frac{h}{3}))k_2. \end{aligned}$$

Integrando en (x_n, x) e imponiendo que $u(x_n) = y_n$, tenemos que

$$u(x) = y_n - \frac{3}{2h} \left.\frac{(x - (x_n + h))^2}{2}\right|_{x=x_n}^{x} k_1 + \frac{3}{2h} \left.\frac{(x - (x_n + \frac{h}{3}))^2}{2}\right|_{x=x_n}^{x} k_2.$$

De aquí deducimos que

$$y_{n+1} = u(x_{n+1}) = y_n + h\left(\frac{3k_1}{4} + \frac{k_2}{4}\right),$$

y que

$$u\left(x_n+\frac{h}{3}\right)=y_n+h\left(\frac{5k_1}{12}-\frac{k_2}{12}\right).$$

En resumen,

$$\begin{cases} k_1=f(x_n+\frac{h}{3},y_n+h\left(\frac{5k_1}{12}-\frac{k_2}{12}\right), \\ k_2=f(x_n+h,y_n+h\left(\frac{3k_1}{4}+\frac{k_2}{4}\right)), \\ y_{n+1}=y_n+h\left(\frac{3k_1}{4}+\frac{k_2}{4}\right). \end{cases}$$

Se trata por tanto de un método de Runge-Kutta de tablero

$$\begin{array}{c|cc} 1/3 & 5/12 & -1/12 \\ 1 & 3/4 & 1/4 \\ \hline & 3/4 & 1/4. \end{array}$$

---

2. Comprobar que la condición (3.13) conduce a la recurrencia (3.14).

Solución. Un sencillo cálculo muestra que

$$\begin{aligned} Q'_{n,2}(x)=&\frac{y_n}{2h^2}\left((x-x_{n+2})+(x-x_{n+1})\right) \\ &-\frac{y_{n+1}}{h^2}\left((x-x_{n+2})+(x-x_n)\right)+\frac{y_{n+2}}{2h^2}\left((x-x_{n+1})+(x-x_n)\right). \end{aligned}$$

Por consiguiente,

$$Q'_{n,2}(x_{n+2})=\frac{1}{h}\left(\frac{3}{2}y_{n+2}-2y_{n+1}+\frac{1}{2}y_n\right),$$

de donde se deduce inmediatamente el resultado.

---

3. Dadas las aproximaciones $y_n,\ldots,y_{n+k-1}$, consideramos el polinomio $Q_{n,k}$ de grado menor o igual que k que interpola a $(x_n,y_n),\ldots,(x_{n+k},y_{n+k})$.

Obviamente este polinomio no se puede construir, porque desconocemos y_{n+k}. Se impone ahora que $Q_{n,k}$ satisfaga la ecuación diferencial en x_{n+k}, esto es

$$Q'_{n,k}(x_{n+k}) = f(x_{n+k}, Q_{n,k}(x_{n+k})) = f(x_{n+k}, y_{n+k}).$$

El método resultante se conoce como *fórmula BDF de k pasos* (del inglés "backward differentiation formulae").

(a) Obtener la recurrencia para la fórmula BDF de k pasos.

(b) Obtener, usando la fórmula para el error de interpolación, una fórmula para el residuo de la fórmula BDF de k pasos que demuestre que $\|R_n\| \leq Ch^{k+1}$.

Solución. (a) El polinomio $Q_{n,k}$ de grado menor o igual que k que interpola a $(x_n, y_n), \ldots, (x_{n+k}, y_{n+k})$ está dado por

$$Q_{n,k}(x) = \sum_{j=0}^{k} y_{n+j} L_j(x), \qquad L_j(x) = \prod_{l=0,\ l\neq j}^{k} \frac{x - x_{n+l}}{x_{n+j} - x_{n+l}}.$$

Definiendo

$$L_j^*(s) = \prod_{l=0,\ l\neq j}^{k} \frac{s-l}{j-l},$$

se tiene que $L_j(x) = L_j^*(\frac{x-x_n}{h})$, y de aquí que

$$Q'_{n,k}(x_{n+k}) = \frac{1}{h} \sum_{j=0}^{k} y_{n+j} (L_j^*)'(k).$$

Usando que $Q'_{n,k}(x_{n+k}) = f(x_{n+k}, y_{n+k})$, se llega a

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h f(x_{n+k}, y_{n+k}), \qquad \alpha_j = (L_j^*)'(k),$$

que una vez normalizada (dividiendo por α_k) es la *fórmula BDF de k pasos*. Es fácil ver que $\alpha_0 \neq 0$; por tanto se tiene un método de k pasos "genuino".

(b) Sea $q_{n,k}(x)$ el polinomio de grado menor o igual que k que interpola los $k+1$ valores exactos

$$(x_n, y(x_n)), \ldots, (x_{n+k}, y(x_{n+k})).$$

Entonces

$$R_n = hq'_{n,k}(x_{n+k}) - hf(x_{n+k}, y(x_{n+k})) = h(q'_{n,k}(x_{n+k}) - y'(x_{n+k})).$$

Sea $g(x) = q_{n,k}(x) - y(x)$. Esta función se anula en los $k+1$ puntos $x_n, \ldots, x_{n+k}$. Por el Teorema de Rolle, $g'(x)$ se anula (componente a componente) en k puntos intermedios a los anteriores. Así $q'_{n,k}(x)$ interpola (componente a componente) a $y'(x)$ en k puntos $\overline{\eta}_l$, $l = 0, \ldots, k-1$. La fórmula para el error de interpolación produce

$$q'_{n,k}(x) - y'(x) = -\frac{y^{(k+1)}(\overline{\xi}(x))}{k!} \prod_{l=0}^{k-1} (x - \overline{\eta}_l).$$

Por tanto

$$\|R_n\| \leq \frac{h \operatorname{máx}_{x\in[a,b]} \|y^{(k+1)}(x)\|}{k!} \prod_{l=0}^{k-1} |x_{n+k} - \overline{\eta}_l| \leq h^{k+1} \max_{x\in[a,b]} \|y^{(k+1)}(x)\|,$$

y concluimos que el método es consistente de orden $p = k$.

Sección 3.5

1. El uso del método de Newton en la Regla del Trapecio requiere evaluar la matriz jacobiana $D_y f$. Esta función debe ser suministrada al integrador. Sin embargo, en las aplicaciones prácticas especificar estas derivadas parciales analíticamente es con frecuencia una tarea difícil o engorrosa. Una alternativa es usar diferencias aproximadas: en $y = y_n^{[k]}$ se evalúan $\hat{f} = f(x_n, \hat{y})$ y $\tilde{f} = f(x_n, \tilde{y})$, donde $\hat{y}$ y $\tilde{y}$ son perturbaciones de y en una coordenada, $\hat{y}_j = y_j + \epsilon$, $\tilde{y}_j = y_j - \epsilon$, $\hat{y}_l = \tilde{y}_l = y_l$, $l \neq j$. Entonces, la j-ésima columna de $D_y f$ se puede aproximar por

$$D_y f \approx \frac{1}{2\epsilon}(\hat{f} - \tilde{f}),$$

siendo ϵ un parámetro positivo pequeño.

Este truco, fácil de programar, suele funcionar bien con la elección $\epsilon = 10^{-\gamma}$, si se trabaja con aritmética de coma flotante con aproximadamente 2γ dígitos significativos (en el caso de Matlab, $\gamma = 7$).

Observación. Esta aproximación de la matriz jacobiana puede resultar costosa en algunos casos y no siempre funciona bien.

(a) Modificar el programa `trapecionw` que se os ha proporcionado de forma que use una aproximación del jacobiano en lugar de la función jacobiana suministrada por el usuario.

(b) Otra alternativa para aproximar el jacobiano numéricamente es usar la función de Matlab `numjac`. Modificar el programa `trapecionw` de manera que aproxime el jacobiano por medio de esta función.

Solución. (a)

```
function [y,evfun]=problema3_5_1a(ld,x,y0,itmax,tol)

N=length(x)-1;
evfun=0;
d=length(y0); % dimension del sistema
I=eye(d);
y(:,1)=y0;
yk=y(:,1);
for n=1:N, % Bucle para recorrer toda la discretizacion
    numit=0;
    errornewton=tol+1;
    hn=x(n+1)-x(n);
    fn=feval(ld,x(n),y(:,n));
    evfun=evfun+1;
    while (numit<itmax && errornewton>tol)
        numit=numit+1;
        g=yk-y(:,n)-.5*hn*(fn+feval(ld,x(n+1),yk));
        evfun=evfun+1;
        % calculo aproximado del jacobiano del lado derecho
        jyld=[];
        for j=1:d
            haty=yk;
            haty(j)=haty(j)+1e-7;
            tildey=yk;
            tildey(j)=tildey(j)-1e-7;
            jyld=[jyld,(feval(ld,x(n+1),haty)-feval(ld,x(n+1),tildey))*.5e7];
            evfun=evfun+2;
        end
        jacg=I-.5*hn*jyld;
        correccion=-jacg\g;
        errornewton=norm(correccion,inf);
        yk=yk+correccion;
    end
    if errornewton<=tol
        y(:,n+1)=yk; %incluyo la ultima correccion
    else
        error('newton no converge')
    end
end
```

(b)

```
function [y,evfun]=problema3_5_1b(ld,x,y0,itmax,tol)

N=length(x)-1;
evfun=0;
fac=[];
d=length(y0); % dimension del sistema
I=eye(d);
y(:,1)=y0;
yk=y(:,1);
for n=1:N, % Bucle para recorrer toda la discretizacion
    numit=0;
    errornewton=tol+1;
    hn=x(n+1)-x(n);
    fn=feval(ld,x(n),y(:,n));
    evfun=evfun+1;
    while (numit<itmax && errornewton>tol)
        numit=numit+1;
        fk=feval(ld,x(n+1),yk);
        evfun=evfun+1;
        g=yk-y(:,n)-.5*hn*(fn+fk);
        [dfdy,fac] = numjac(ld,x(n+1),yk,fk,tol*ones(d,1),fac,0);
        evfun=evfun+2*d;
        % suponemos el mismo coste que en nuestra aproximación
        % numérica del jacobiano
        jacg=I-.5*hn*dfdy;
        correccion=-jacg\g;
        errornewton=norm(correccion,inf);
        yk=yk+correccion;
    end
    if errornewton<=tol
        y(:,n+1)=yk; %incluyo la ultima correccion
    else
        error('newton no converge')
    end
end
```

---

2. El programa `trapeciopf` que se os ha suministrado estima el error en la iteración de punto fijo como la diferencia entre las dos últimas iteraciones. Esta estimación con frecuencia sobreestima o subestima el error. Un criterio mejor es tomar

$$\frac{\lambda}{1-\lambda}\|y_n^{[k+1]} - y_n^{[k]}\| \leq \texttt{TOL},$$

donde `TOL` es la tolerancia para la iteración de punto fijo proporcionada por el usuario y λ es una medida de la velocidad de convergencia de la

iteración, que se puede estimar por

$$\lambda = \left(\frac{\|y_n^{[k+1]} - y_n^{[k]}\|}{\|y_n^{[1]} - y_n^{[0]}\|} \right)^{1/k}.$$

(a) Modificar el programa `trapeciopf` de manera que use esta estimación mejorada del error de punto fijo.

(b) Elaborar un diagrama que mida la eficiencia de ambos métodos (el original y el modificado) al aplicarlos al problema del péndulo.

Solución. (a)

```
function [y,evfun]=problema3_5_2a(ld,x,y0,itmax,tol)

N=length(x)-1;
evfun=0;
y(:,1)=y0;
for n=1:N, % Bucle para recorrer toda la discretizacion
    numit=0;
    errorpf=tol+1;
    hn=(x(n+1)-x(n));
    fn=feval(ld,x(n),y(:,n));
    evfun=evfun+1;
    yk=y(:,n)+hn*fn;
    y1menosy0=.5*hn*(-fn+feval(ld,x(n+1),yk));
    numit=numit+1;
    yk=yk+y1menosy0;
    y1menosy0=norm(y1menosy0,inf);
    while (numit<itmax && errorpf>tol)
        numit=numit+1;
        ykmas1=y(:,n)+.5*hn*(fn+feval(ld,x(n+1),yk));
        evfun=evfun+1;
        lambda=(norm(ykmas1-yk,inf)/(y1menosy0))^(1/(numit-1));
        errorpf=norm(lambda*(ykmas1-yk)/(1-lambda),inf);
        yk=ykmas1;
    end
    if errorpf<=tol
        y(:,n+1)=yk; %incluyo la ultima trapeciopfej35.mcorreccion
    else
        error('la iteracion de punto fijo no converge')
    end
end
```

(b) Para elaborar el diagrama pedido usamos el siguiente programa:

```
function problema3_5_2b

close all, clear all

% Calculo de la solucion .exacta"
opciones=odeset('AbsTol',1e-20,'RelTol',2.22045e-014);
solpend=ode45(@pend,[0,1],[1;1],opciones);

pd=12; % Numero de puntos en el diagrama
emetodo=zeros(1,pd); fmetodo=zeros(1,pd);

% Regla del trapecio con una estimacion burda del error
N=10;
for i=1:pd
    N=2*N;
    x=linspace(0,1,N+1);
    [y,f]=trapecio_pf(@pend,x,[1;1],15,1e-14);
    solexacta=deval(solpend,x);
    emetodo(i)=max(max(abs(solexacta-y)));
    fmetodo(i)=f;
end
loglog(fmetodo,emetodo,'r-*')
hold

% Regla del trapecio con una estimacion mejor del error
N=10;
for i=1:pd
    N=2*N;
    x=linspace(0,1,N+1);
    [y,f]=problema3_5_2a(@pend,x,[1;1],15,1e-14);
    solexacta=deval(solpend,x);
    emetodo(i)=max(max(abs(solexacta-y)));
    fmetodo(i)=f;
end
loglog(fmetodo,emetodo,'b-o')
xlabel('evaluaciones de funcion'), label('error')
legend('Estimaci\'{o}n burda','Estimaci\'{o}n fina')

function yprima = pend(t,y)
%PEND Pendulo simple

yprima = [y(2); -sin(y(1))];
```

El resultado es el diagrama que se muestra en la Figura 1. Se observa que hacer las cosas bien vale la pena: hay un cierto ahorro.

---

3. Las evaluaciones del jacobiano en el método de Newton pueden ser costosas, y también lo es la resolución de los sistemas lineales resultantes. Por ello, se suele usar un método de Newton modificado, que mantiene fijo el jacobiano, $D_y f \approx D_y f(x_{n+1}, y_{n+1}^{[0]})$. De esta manera sólo se evalúa

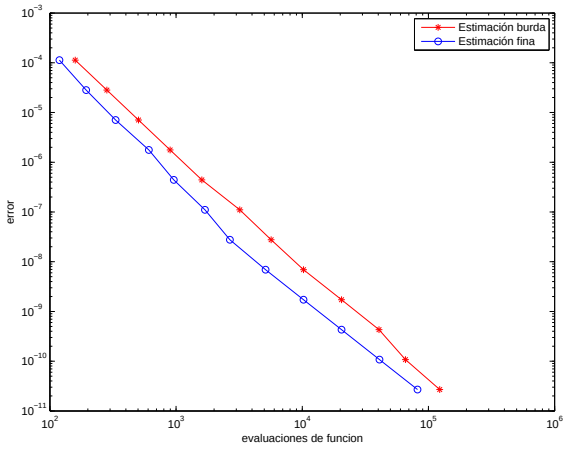

Figura 1: Eficiencia regla del trapecio con dos estimaciones diferentes del error en la iteración de punto fijo.

el jacobiano una vez. Y lo que es aún mejor, se hace una (única) descomposición LU (lo que conlleva $\frac{d^3}{3} + O(d^2)$ operaciones), que se emplea en todos los pasos de la iteración para resolver los sistemas lineales (el coste es $O(d^2)$).

(a) Modificar el programa `trapecionw` de forma que recoja las ideas que acabamos de explicar.

(b) Elaborar un diagrama que mida la eficiencia de ambos métodos (el original y el modificado) al aplicarlos al problema del péndulo.

Solución. (a)

```
function [y,evf,evj]=problema3_5_3a(ld,jyld,x,y0,itm,tol)
N=length(x)-1; evf=0; evj=0; I=eye(length(y0));
y(:,1)=y0;
for n=1:N, % Bucle para recorrer la discretizacion
    numit=0;
    enewton=tol+1;
    hn=x(n+1)-x(n);
    fn=feval(ld,x(n),y(:,n));
    evf=evf+1;
    yk=y(:,n)+hn*fn;
    jacg=I-.5*hn*feval(jyld,x(n+1),yk);
    evj=evj+1;
    [L,U,P]=lu(jacg); % PA=LU, P mat. permut., L trian. inf., U trian. sup.
    while (numit<itm && enewton>tol) % Bucle del Newton
        numit=numit+1;
        % g es la funcion de la que hay que encontrar el cero
        g=yk-y(:,n)-.5*hn*(fn+feval(ld,x(n+1),yk));
        evf=evf+1;
        b=-P*g;
        Uc=L\b;
        correccion=U\Uc;
        enewton=norm(correccion,inf);
        yk=yk+correccion;
    end
    if enewton<=tol
        y(:,n+1)=yk; %incluyo la ultima correccion
    else
        error('newton no converge')
    end
end
```

(b) El programa para realizar el diagrama de eficiencia pedido es muy similar al del problema 3.5.2(b). El diagrama se muestra en la Figura 2. Se observa que congelar el jacobiano supone un cierto ahorro.

---

4. Si integramos la EDO $y'(x) = f(x, y(x))$ en el intervalo (x_n, x_{n+1}) se tiene que $y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f(s, y(s))\, ds$. Si aproximamos el integrando por medio de la regla rectangular derecha, llegamos al llamado *método de Euler implícito*,

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}).$$

Programar una función de Matlab que calcule soluciones numéricas para sistemas por este método. Los sistemas no lineales resultantes se resolverán:

(a) por iteración de punto fijo;

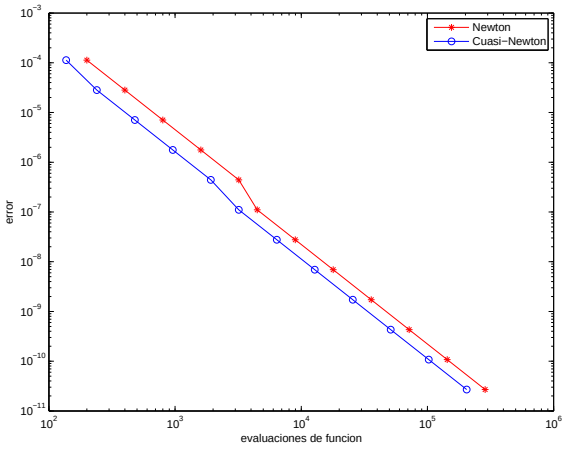

Figura 2: Eficiencia regla del trapecio con Newton y con cuasi-Newton.

(b) por el método de Newton.

La estructura de las variables de entrada y salida será la misma que la de las funciones `trapeciop_f` y `trapecio_new` que se os han suministrado.

Solución. (a)

```
function [y,evfun]=problema3_5_4a(ld,x,y0,itmax,tol)

N=length(x)-1;
evfun=0;
y(:,1)=y0;
for n=1:N, % Bucle para recorrer toda la discretizacion
    numit=0;
    errorpf=tol+1;
    hn=(x(n+1)-x(n));
    yk=y(:,n);
    while (numit<itmax && errorpf>tol)
        numit=numit+1;
        ykmas1=y(:,n)+hn*feval(ld,x(n+1),yk);
        evfun=evfun+1;
        errorpf=norm(ykmas1-yk,inf);
        yk=ykmas1;
    end
    if errorpf<=tol
        y(:,n+1)=yk; %incluyo la ultima correccion
    else
        error('la iteracion de punto fijo no converge')
    end
end
```

(b)

```
function [y,evfun,evjac]=problema3_5_4b(ld,jyld,x,y0,itmax,tol)

N=length(x)-1;
evfun=0; evjac=0;
d=length(y0); % dimension del sistema
I=eye(d);
y(:,1)=y0;
yk=y(:,1);
for n=1:N, % Bucle para recorrer toda la discretizacion
    numit=0;
    errornewton=tol+1;
    hn=x(n+1)-x(n);
    while (numit<itmax && errornewton>tol)
        numit=numit+1;
        g=yk-y(:,n)-hn*feval(ld,x(n+1),yk);
        evfun=evfun+1;
        jacg=I-hn*feval(jyld,x(n+1),yk);
        evjac=evjac+1;
        correccion=-jacg\g;
        errornewton=norm(correccion,inf);
        yk=yk+correccion;
    end
    if errornewton<=tol
        y(:,n+1)=yk; %incluyo la ultima correccion
    else
        error('newton no converge')
    end
end
```

5. Repetir el ejercicio anterior para el método de Runge-Kutta implícito de tablero

$$\begin{array}{c|c} 1/2 & 1/2 \\ \hline & 1. \end{array}$$

Los nombres de los programas serán `rkipf` y `rkinw`.

Solución. (a)

```
function [y,evfun]=problema3_5_5a(ld,x,y0,itmax,tol)

N=length(x)-1;
evfun=0;
y(:,1)=y0;
for n=1:N, % Bucle para recorrer toda la discretizacion
    numit=0;
    errorpf=tol+1;
    h=(x(n+1)-x(n));
    k1=feval(ld,x(n),y(:,n));
    evfun=evfun+1;
    while (numit<itmax & errorpf>tol)
        numit=numit+1;
        nuevok1=feval(ld,x(n)+h/2,y(:,n)+h*k1/2);
        evfun=evfun+1;
        errorpf=norm(nuevok1-k1,inf);
        k1=nuevok1;
    end
    if errorpf<=tol
        y(:,n+1)=y(:,n)+h*k1;
    else
        error('la iteracion de punto fijo no converge')
    end
end
```

(b)

```
function [y,evfun,evjac]=problema3_5_5b(ld,jyld,x,y0,itmax,tol)

N=length(x)-1;
evfun=0;
evjac=0;
d=length(y0); % dimension del sistema
I=eye(d);
y(:,1)=y0;
for n=1:N, % Bucle para recorrer toda la discretizacion
    numit=0;
    errornewton=tol+1;
    h=x(n+1)-x(n);
    k1=feval(ld,x(n),y(:,n));
    evfun=evfun+1;
    while (numit<itmax && errornewton>tol)
        numit=numit+1;
        g=k1-feval(ld,x(n)+h/2,y(:,n)+h*k1/2);
        evfun=evfun+1;
        jacg=I-.5*h*feval(jyld,x(n)+h/2,y(:,n)+h*k1/2);
        evjac=evjac+1;
        correccion=-jacg\g;
        errornewton=norm(correccion,inf);
        k1=k1+correccion;
    end
    if errornewton<=tol
        y(:,n+1)=y(:,n)+h*k1; %incluyo la ultima correccion
    else
        error('newton no converge')
    end
end
```

---

6. Para dibujar un círculo de radio r en una pantalla gráfica, uno puede evaluar pares $x = r\cos\theta$, $y = r\sin\theta$ para una sucesión de valores de θ. Pero esto es costoso. Un método alternativo más barato consiste en considerar el PVI

$$x'(\theta) = -y(\theta), \quad y'(\theta) = x(\theta), \qquad x(0) = r, \quad y(0) = 0,$$

y aproximarlo utilizando algún método numérico. Sin embargo, hay que asegurarse de que la solución obtenida tiene el aspecto deseado.

Aplicar el método de Euler, el método de Euler implícito y la Regla del trapecio al problema, con paso $h = 0{,}02$, para $0 \leq \theta \leq 120$. Determinar si la solución forma una espiral hacia fuera, una espiral hacia dentro o, como se desea, un círculo aproximado. Explicar los resultados observados.

Sugerencia. Esto tiene que ver con una cierta función invariante de x e y, más que con el orden de los métodos.

Solución. Las aproximaciones de la curva solución que comienza en $(1,0)^T$ por medio del método de Euler, del método de Euler implícito y de la regla del trapecio se dan, respectivamente, en las figuras 3, 4 y 5.

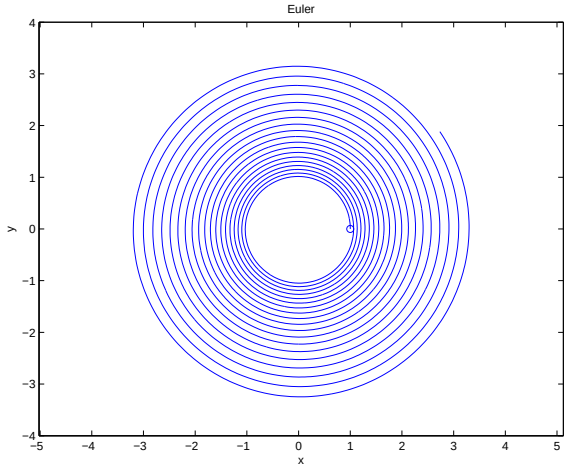

Figura 3: Aproximación por Euler de la solución que empieza en $(1,0)^T$. La aproximación describe una espiral que se abre.

Explicación. La función cuadrática

$$f(\theta) = x^2(\theta) + y^2(\theta)$$

es constante en θ. En efecto, usando el sistema de EDOs satisfechas por x e y se obtiene fácilmente que

$$f'(\theta) = 2x(\theta)x'(\theta) + 2y(\theta)y'(\theta) = -2x(\theta)y(\theta) + 2y(\theta)x(\theta) = 0.$$

Veamos si los métodos que hemos usando preservan este invariante. Usamos una malla $\{\theta_n\}_{n=0}^N$.

La solución numérica producida por el método de Euler viene dada por la recurrencia

$$x_{n+1} = x_n - hy_n, \qquad y_{n+1} = y_n + hx_n.$$

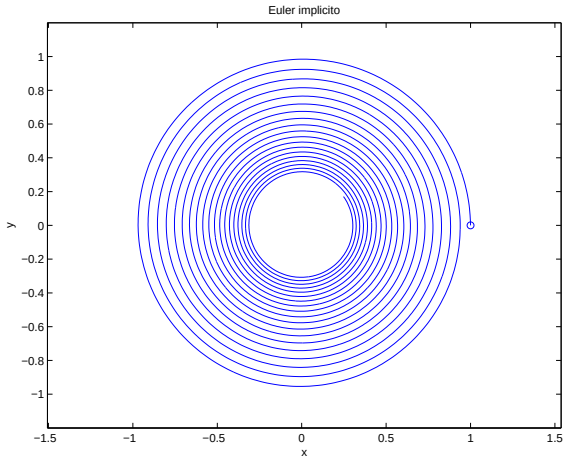

Figura 4: Aproximación por Euler implícito de la solución que empieza en $(1,0)^T$. La aproximación describe una espiral que se cierra.

Así pues,

$$\begin{aligned} x_{n+1}^2 + y_{n+1}^2 &= x_n^2 - 2hx_ny_n + h^2y_n^2 + y_n^2 + 2hx_ny_n + h^2x_n^2 \\ &= (1+h^2)(x_n^2 + y_n^2). \end{aligned}$$

Por consiguiente,

$$x_n^2 + y_n^2 = (1+h^2)^n(x_0^2 + y_0^2),$$

y las soluciones numéricas obtenidas por este método describen, tal y como se observa en el experimento, una espiral que se abre. En la Figura 6 se representa $\log(x_n^2 + y_n^2)$ frente a n. Se comprueba que, como era de esperar se obtiene una recta con pendiente $\log(1 + 0{,}02^2)$.

La solución numérica producida por el método de Euler implícito viene dada por la recurrencia

$$x_{n+1} = x_n - hy_{n+1}, \qquad y_{n+1} = y_n + hx_{n+1}.$$

Así pues,

$$x_{n+1}^2 + y_{n+1}^2 = x_n^2 - 2hx_ny_{n+1} + h^2y_{n+1}^2 + y_n^2 + 2hx_{n+1}y_n + h^2x_{n+1}^2.$$

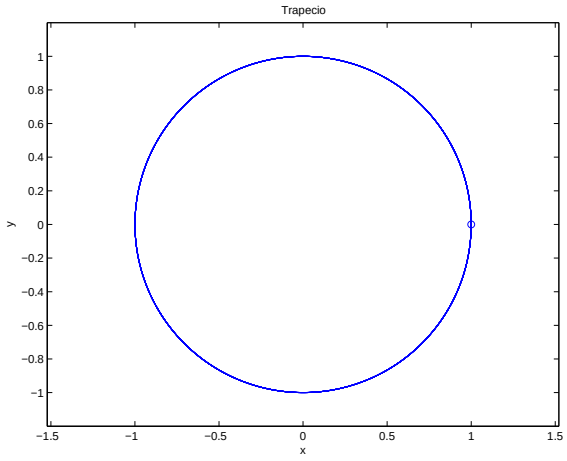

Figura 5: Aproximación por la regla del trapecio de la solución que empieza en $(1,0)^T$. La aproximación está, como la propia solución, en la circunferencia de radio 1.

Para eliminar x_n e y_n en esta expresión usamos la recurrencia del método,

$$x_n = x_{n+1} + hy_{n+1}, \qquad x_n = y_{n+1} - hx_{n+1}.$$

Por consiguiente,

$$\begin{aligned} x_{n+1}^2 + y_{n+1}^2 &= x_n^2 + y_n^2 + h^2(x_{n+1}^2 + y_{n+1}^2) \\ &\quad -2h(x_{n+1} + hy_{n+1})y_{n+1} + 2h(y_{n+1} - hx_{n+1})x_{n+1} \\ &= x_n^2 + y_n^2 - h^2(x_{n+1}^2 + y_{n+1}^2). \end{aligned}$$

Concluimos que

$$x_{n+1}^2 + y_{n+1}^2 = \frac{1}{1+h^2}(x_n^2 + y_n^2),$$

y por tanto que

$$x_n^2 + y_n^2 = \left(\frac{1}{1+h^2}\right)^n (x_0^2 + y_0^2)$$

y las soluciones numéricas obtenidas por este método describen, tal y como se observa en el experimento, una espiral que se cierra. En

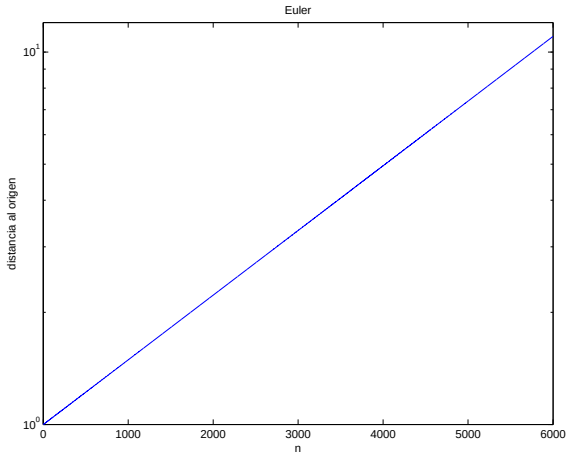

Figura 6: Tasa de crecimiento del radio en la aproximación por el método de Euler.

la Figura 6 se representa $\log(x_n^2 + y_n^2)$ frente a n. Se comprueba que, de acuerdo con lo que acabamos de explicar, se obtiene una recta con pendiente $-\log(1 + 0{,}02^2)$.

La solución numérica producida por la regla del trapecio viene dada por la recurrencia

$$x_{n+1} = x_n - \frac{h}{2}(y_n + y_{n+1}), \qquad y_{n+1} = y_n + \frac{h}{2}(x_n + x_{n+1}).$$

Así pues,

$$\begin{aligned} x_{n+1}^2 + y_{n+1}^2 &= x_n^2 + \tfrac{h^2}{4}y_n^2 + \tfrac{h^2}{4}y_{n+1}^2 - hx_ny_n - hx_ny_{n+1} + \tfrac{h^2}{2}y_ny_{n+1} \\ &\quad + y_n^2 + \tfrac{h^2}{4}x_n^2 + \tfrac{h^2}{4}x_{n+1}^2 + hx_ny_n + hx_ny_{n+1} + \tfrac{h^2}{2}x_nx_{n+1} \\ &= (x_n^2 + y_n^2)\left(1 + \tfrac{h^2}{4}\right) + \tfrac{h^2}{4}(x_{n+1}^2 + y_{n+1}^2) \\ &\quad - h(x_{n+1} + \tfrac{h}{2}(y_n + y_{n+1}))y_{n+1} + \tfrac{h^2}{2}y_ny_{n+1} \\ &\quad + h(y_{n+1} - \tfrac{h}{2}(x_n + x_{n+1}))x_{n+1} + \tfrac{h^2}{2}x_nx_{n+1} \\ &= (x_n^2 + y_n^2)\left(1 + \tfrac{h^2}{4}\right) - \tfrac{h^2}{4}(x_{n+1}^2 + y_{n+1}^2). \end{aligned}$$

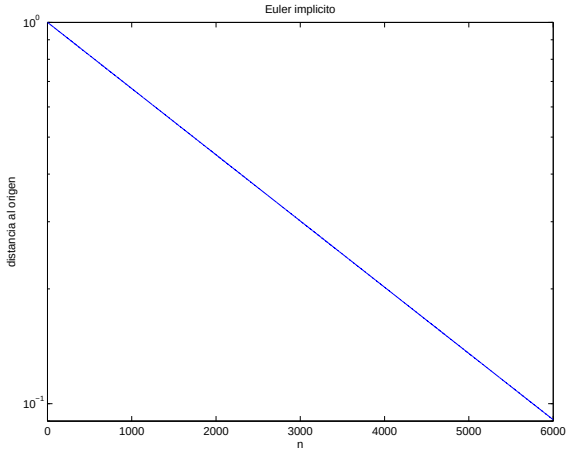

Figura 7: Tasa de decrecimiento del radio en la aproximación por el método de Euler implicito.

Concluimos que

$$x_{n+1}^2 + y_{n+1}^2 = x_n^2 + y_n^2.$$

Así pues, la regla del trapecio respeta el invariante del movimiento, y las soluciones numéricas obtenidas por este método están, tal y como se observa en el experimento, sobre una circunferencia.

---

7. El sistema de EDOs

$$(y^1)' = \alpha - y^1 - \frac{4y^1y^2}{1+(y^1)^2}, \quad (y^2)' = \beta y^1\left(1 - \frac{y^2}{1+(y^1)^2}\right),$$

donde α y β son parámetros, representa de forma simplificada a cierta reacción química. Para cada valor del parámetro α hay un valor crítico del parámetro β, $\beta_c = \frac{3\alpha}{5} - \frac{25}{\alpha}$, tal que para $\beta > \beta_c$ las trayectorias de las soluciones decaen en amplitud y se acercan en el plano de fases en forma espiral a un punto fijo estable, mientras que para $\beta < \beta_c$ las trayectorias oscilan sin amortiguarse y se ven atraídas por un ciclo límite estable (esto es lo que se conoce como una bifurcación de Hopf).

   (a) Fijamos $\alpha = 10$. Usar cualquiera de los métodos que se han programado hasta el momento con longitud de paso fija $h = 0{,}01$ para

aproximar la solución del problema que empieza en $y^1(0) = 0$, $y^2(0) = 2$, para $t \in [0, 20]$, con $\beta = 2$ y $\beta = 4$. En cada caso dibujar y^1 e y^2 frente a t. Describir lo observado.

(b) Investigar cuál es la situación cerca del valor crítico $\beta_c = 3{,}5$ (quizá convenga incrementar la longitud del intervalo de integración para ver mejor lo que sucede).

Solución. (a) Hemos realizado los cálculos con la regla del trapecio (punto fijo). Los resultados se muestran en las figuras 8 y 9. Se observa que con $\beta = 2$ la curva solución se aproxima muy rápidamente a un ciclo límite (que corresponde a una solución periódica, mientras que con $\beta = 4$ converge velozmente a un punto crítico (una solución estacionaria).

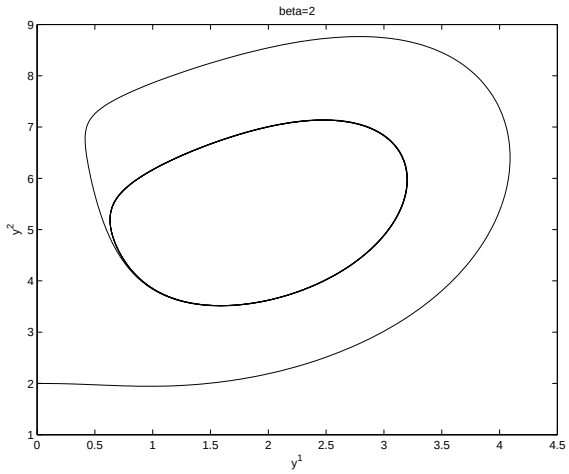

Figura 8: $\beta = 2$: convergencia rápida a un ciclo límite.

(b) Hemos ampliado el tiempo de integración para cubrir el intervalo $[0, 50]$, y hemos considerado los valores $\beta = 3{,}4$ y $\beta = 3{,}6$. En el primer caso, Figura 10, se tiene convergencia a un ciclo límite, pero muchísimo más lenta que con $\beta = 2$. En el segundo, Figura 11, se converge a un punto crítico, pero también de forma mucho más lenta. El valor crítico $\beta = 3{,}5$, marca la desaparición del ciclo límite.

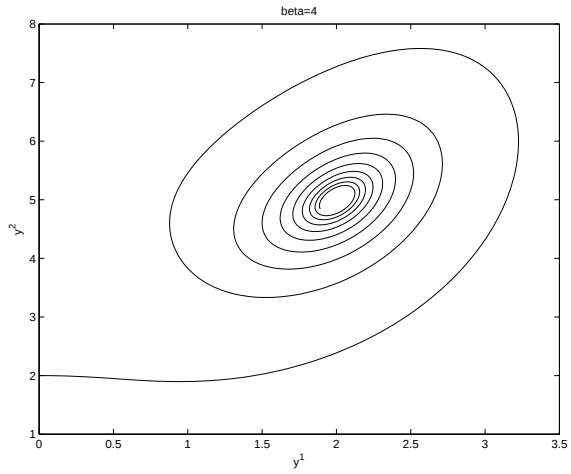

Figura 9: $\beta = 4$: convergencia rápida a un punto crítico.

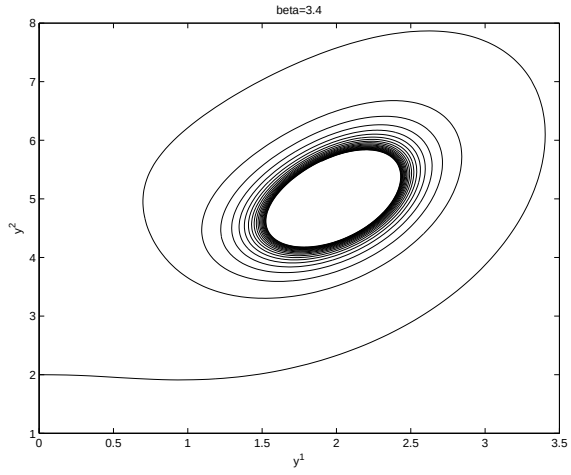

Figura 10: $\beta = 3{,}4$: convergencia lenta a un ciclo límite.

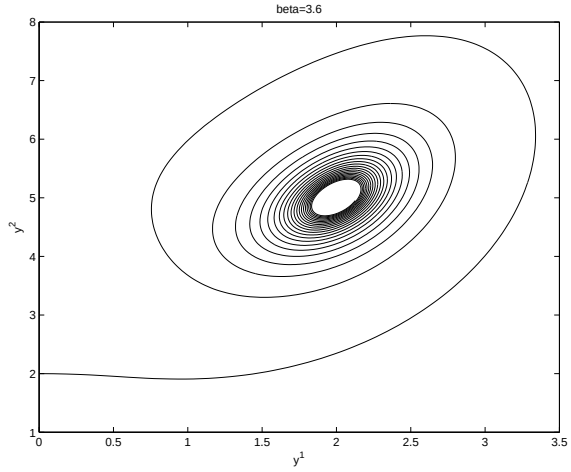

Figura 11: $\beta = 3{,}6$: convergencia lenta a un punto crítico.