UNIVERSIDAD DE CONCEPCION
FACULTAD DE CIENCIAS
FISICAS Y MATEMATICAS
DEPARTAMENTO DE INGENIERIA MATEMATICA

Cálculo Numérico(521230/525240)
Laboratorio 6

Problemas de Valores Iniciales: Parte II

Ejercicios Laboratorio

Actividad 1: (Desarrollar en laboratorio por el/la ayudante)

Considere el problema de valores iniciales

$$x'(t) = -3t\,(x(t))^2 + \frac{1}{1+t^3}, \qquad t \in [0,\,5],$$
$$x(0) = 0.$$

Escriba un rutero en el que:

1.1 Defina una función para evaluar la parte derecha de la ecuación diferencial anterior.

1.2 Resuelva el problema de valores iniciales con el método de Euler explícito y $n = 10$, $n = 20$, $n = 40$ y $n = 80$.

1.3 Grafique las cuatro aproximaciones obtenidas, utilice un color distinto para cada una.

1.4 Calcule la diferencia entre las aproximaciones a $x(t)$ calculadas con los pares de valores
- $n = 10$ y $n = 20$,
- $n = 20$ y $n = 40$,
- $n = 40$ y $n = 80$.

Continúe leyendo para entender qué estamos pidiendo exactamente. Tomemos, por ejemplo, las aproximaciones calculadas con $n = 10$ y $n = 20$. Si `t1,x1` es la salida de Euler explícito con $n = 10$ y `t2,x2`, la salida con $n = 20$, entonces `t1` y `x1` son vectores con 11 componentes, mientras que `t2` y `x2` tienen 21 componentes cada uno.

Si solamente tomamos los valores en las posiciones impares de `t2`, el vector resultante es igual a `t1`.

Es decir, si escribimos el siguiente par de comandos en MATLAB: `j=0:20; t21 = t2(2*j+1);` los vectores `t21` y `t1` son iguales.

Si ahora hacemos `x21 = x2(2*j+1);` los vectores `x1` y `x21` son aproximaciones a $x($`t1`$)$, las aproximaciones en `x1` se calcularon con $n = 10$ y las aproximaciones en `x21`, con $n = 20$.

Cuando pedimos calcular la diferencia entre las aproximaciones con $n = 10$ y con $n = 20$, nos referimos a calcular la mayor diferencia entre `x1` y `x21`. Esto lo logramos con el comando MATLAB: `max(abs(x1-x21))`.

Podemos averiguar incluso en qué valor de t ocurre la mayor diferencia entre `x1` y `x21`. Para ello podemos escribir en MATLAB, en lugar de `max(abs(x1-x21))`, el comando:

`[m,id] = max(abs(x1-x21));` Así, `m` contiene la mayor distancia entre `x1` y `21` y `t1(id)`, el valor de t donde se diferencian más ambas aproximaciones.

Repita lo explicado antes, pero con los pares de aproximaciones calculadas con $n = 20$, $n = 40$ y $n = 40$, $n = 80$. Tenga presente que los vectores resultantes no tienen 11 y 21 componentes, sino 21 y 41 y 41 y 81 respectivamente.

Actividad evaluada en Canvas: Las preguntas en la actividad evaluada son relativas a la actividad 1 de esta guía.

Actividad 2: (Desarrollar en laboratorio por el/la ayudante)

Considere el problema de valores iniciales

$$\begin{aligned} y'(t) &= -200y(t) + 200\sin(t) + \cos(t), \qquad t \in [0,1], \\ y(0) &= 1; \end{aligned}$$

2.1 Descargue el archivo `euler_implicito` del módulo *Capítulo 3: Ecuaciones diferenciales ordinarias* de la página Canvas del curso.

2.2 Resuelva, con `euler_implicito` y `euler` y con $n = 80$ el problema anterior.

2.3 Grafique, en ventanas distintas, las aproximaciones obtenidas con ambos métodos. ¿Son similares los gráficos resultantes?

2.4 Calcule, con el método de Euler explícito y $n = 160$ una nueva aproximación a la solución exacta del PVI. Grafíquela. ¿Es ésta similar a la aproximación calculada con el método de Euler implícito y $n = 80$?

2.5 Repita la pregunta anterior con $n = 320$. ¿Es ahora la aproximación que retorna Euler explícito similar a la obtenida con Euler implícito y $n = 80$?

2.6 ¿Contradicen los resultados obtenidos el hecho de que los métodos de Euler sean ambos métodos convergentes?

2.7 La semana pasada utilizamos el método `ode45` de MATLAB para resolver un PVI. Utilícelo para resolver este problema, ¿en cuántos valores de $t \in [0,1]$ se calculan aproximaciones a $y(t)$ con este método?

2.8 Resuelva el PVI con `ode15s` (lea comentario más abajo). ¿En cuántos valores de $t \in [0,1]$ se calculan aproximaciones a $y(t)$ con este método?

Observación: El problema en esta actividad es un ejemplo de problema stiff. Con Euler explícito puede calcularse una buena aproximación a la solución exacta de este problema, pero dividiendo el intervalo de integración en muchos más subintervalos que los que necesita Euler implícito para lograr la misma precisión.

Los métodos MATLAB `ode45` y `ode15s` se diferencian en que el primero no es adecuado para problemas stiff y el segundo sí. La diferencia en la cantidad de aproximaciones que deben calcularse con ambos para resolver de manera satisfactoria este problema te indica que estás, efectivamente, en presencia de un problema stiff.

Actividad 3: (Desarrollar como trabajo independiente)

Complete la actividad 1, pero haga ahora un ciclo en el que calcule aproximaciones a la solución exacta del PVI con el método de Euler explícito y $n = 10, 20, 40, \ldots$, es decir, duplicando cada vez la cantidad de subintervalos hasta que la diferencia entre dos aproximaciones sucesivas (calculada como en la actividad 1) sea menor o igual que 10^{-4}.

Para resolver el problema de valores iniciales $y'(t) = f(t, y(t))$, $t \in [a, b]$, $y(a) = y_a$, siendo $y : [a, b] \to \mathbb{R}$, con el comando `ode15s` debe procederse de la manera siguiente:

1. escribir una función para evaluar $f(t, y(t))$. Esta función debe, dados números reales t e y, retornar el valor de $f(t, y)$. Esta función tiene la misma sintaxis que las que escribimos para resolver los problemas en las actividades 1 y 2.

2. llamar a `ode15s` dando como parámetros de entrada: función para evaluar $f(t, y(t))$, vector con dos componentes de la forma `[a,b]`, valor inicial y_a. A diferencia del llamado a Euler explícito, ahora entregamos los valores de a y b agrupados en un vector y n (cantidad de subintervalos en los que dividir $[a, b]$) no es un parámetro de entrada.

   Si el llamado es de la forma `[tv,yv] = ode15s(mif,[a,b],ya)`, `tv` e `yv` son vectores con estas propiedades: en $tv = [t_1, t_2, t_3, \ldots, t_n]$ se almacenan los valores, entre a y b, en los que `ode45` retorna una aproximación a $y(t_i)$, $i = 1, 2, \ldots, n$. El vector $yv = [y_1, y_2, y_3, \ldots, y_n]$ contiene las aproximaciones y_i a $y(t_i)$.