

Parcial 1

José David Ruiz Álvarez

josed.ruiz@udea.edu.co

Instituto de Física, Facultad de Ciencias Exactas y Naturales

Universidad de Antioquia

26 de septiembre de 2019

1. Contenido

Los siguientes problemas son para ser desarrollado en grupo de máximo dos personas.

2. Problema 1: Obligatoriamente en C++

Considere como ejemplo el juego de la vida con sus reglas tradicionales (ver Reglas). Ahora, un juego de la vida de 2 especies que compiten, para ello considere una matriz cuadrada de dimensiones $n \times n$ en donde cada uno de sus elementos puede asumir dos valores 0, 1 o 2. El 1 representa una celda donde hay una célula viva de la especie 1 y el 2 representa una celda donde hay una célula viva de la especie 2, el 0 representa una celda donde no hay células vivas. Dicha matriz evoluciona temporalmente a través de iteraciones de acuerdo a las siguiente reglas:

- Si un elemento tiene como valor 0 y tiene 3 vecinos con valor 1 y dos o más vecinos con valor 0, transforma su valor a 1.
- Si un elemento tiene como valor 0 y tiene 4 vecinos con valor 2 y uno o más vecinos con valor 1, transforma su valor a 1.
- Si un elemento tiene como valor 1 y tiene 2 o 3 vecinos con valor 1, conserva su valor de 1.
- Si un elemento tiene como valor 2 y tiene 2 o 3 vecinos con valor 2, conserva su valor de 2.
- Si un elemento tiene como valor 1 y tiene al menos 4 vecinos con valor 2 y máximo un vecino 1 vecino con valor 1. transforma su valor a 2.
- Si un elemento tiene como valor 2 y tiene 4 o 5 vecinos con valor 1 y máximo un vecino 2 vecino con valor 2. transforma su valor a 1.

- Si un elemento tiene como valor 1 y tiene al menos 5 vecinos valor 1, transforma su valor a 0.
- Si un elemento tiene como valor 2 y tiene 5 vecinos valor 2, transforma su valor a 0.

Desarrolle un script que evolucione el sistema del juego de la vida con la reglas descritas.

- El script para que recibir por entrada de línea de comandos el valor n que fija las dimensiones de la matriz. Sólo considere matrices cuadradas. (Valor=0.5)
- Defina una clase que contenga la matriz y que tenga métodos que la hagan evolucionar de acuerdo a las reglas estipuladas. (Valor=1.5)
- Haga evolucionar el sistema en al menos 1000 iteraciones e imprima en pantalla cada estado de forma tal que se vea evolucionar el sistema de forma estacionaria (que se sobrescriba cada uno de los elementos en cada iteración) (Valor=0.5)

Entregables problema 1: Script en C++ más un script en Bash que compile y corra el script de C++. Sólo se pueden utilizar las librerías estándar de C, arrays y vectores.

3. Problema 2: Obligatoriamente en C++ o Python

Considere un sistema de dos gases que sólo interactúan entre ellos. Dichos gases interactúan entre ellos de forma tal que la población de partículas del gas 1 n_1 y del gas 2 n_2 evolucionan con el tiempo de acuerdo a las siguientes ecuaciones diferenciales acopladas:

$$n_2' = -n_2(c - d \times n_1); \quad n_1' = n_1(a - b \times n_2) \quad (1)$$

donde se entiende que $n_i' = \frac{dn_i}{dt}$ y a, b, c, d son números reales.

Para un sistema de dos ecuaciones diferenciales acopladas, $y'(t) = f(t, x(t), y(t)); \quad x'(t) = g(t, x(t), y(t))$ podemos generalizar el método de Euler usando las siguientes ecuaciones:

$$y_{n+1} = y_n + hf(t_n, x_n, y_n); \quad x_{n+1} = x_n + hg(t_n, x_n, y_n) \quad (2)$$

con $h = (b - a)/N = t_{n+1} - t_n$, $x_n = x(t_n)$ y $y_n = y(t_n)$.

Con definiciones similares se puede también generalizar el método de Runge-Kutta de orden 4.

Solucione cada uno de los siguientes problemas:

- Implemente (escriba un script) el método de Euler y de RK4 para el sistema de dos ecuaciones diferenciales ordinarias acopladas. (Valor=0.5)

- Utilice ambos métodos y la librería odeint de python para solucionar el sistema de ecuaciones descrito en la ecuación (1) con $a = b = c = d = 1$ y condiciones iniciales $n_1(0) = 1,50$, $n_2(0) = 1,00$. Encuentre la solución al sistema para $t = [0, 12]$. Grafique las soluciones en los planos (t, n_1) , (t, n_2) y (n_1, n_2) (Valor=1.0)
- Utilizando la solución encontrada con el método odeint como si fuese la solución exacta muestre si las soluciones encontradas con el método de Euler y RK4 son o no convergentes. (Valor=1.0)

Entregables problema 2: Scripts completos, gráficos solicitados. Script en Bash que ejecute los scripts desarrollados.

Sobrepuntaje global: Se dará un sobrepuntaje global de 0.5 si TODOS los scripts de los dos problemas están exhaustivamente comentados y explicados.

Método de entrega: Pull request al repositorio central del curso adentro de la carpeta llamada “Parcial1” con todos los entregables para solo uno de los estudiantes en la carpeta de su número de cédula respectivo. Debe además incluir en el pull request los nombres y números de cédulas de los dos integrantes del grupo.