Tarea 1

Seminario de Geometría B

Grupos Kleinianos: Geometría y Visualización

Renato Leriche Vázquez y José Daniel Blancas Camarena

Semestre 2023 - 2

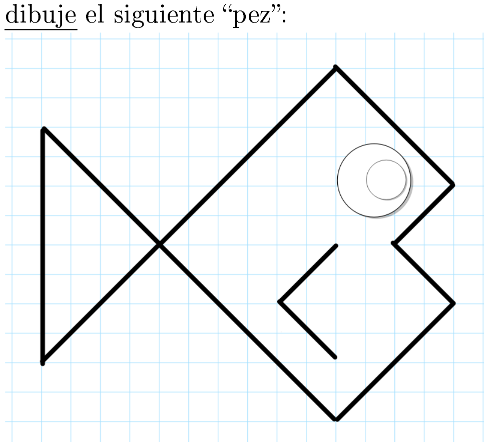
Programación

Notas: En la programación de *functions* tienes que decidir cuáles son los argumentos y valores que regresan adecuados. Las implementaciones deben ser en el lenguaje en ***Julia***. Los dibujos deben hacerse utilizando el paquete ***Luxor***.

1. **[2 Pts]**. Mejora el algoritmo que determina si un número entero es primo (visto en clase), donde no se tomen como "presuntos divisores" los múltiplos de 2 mayores que 2, e impleméntalo.

2. **[1 Pt]**. Programa una function que implemente el método de Newton-Raphson (visto en clase) para encontrar una raíz de funciones $f : \mathbb{R} \to \mathbb{R}$.

3. **[2 Pts]**. Programa una *function* que, a partir de una lista de puntos, dibuje el siguiente "pez":



(Ayuda: Usa la misma idea del dibujo para el Dr. Stickler visto en clase.)

4. **[2 Pts]**. Programa una *function* para dibujar la gráfica de cualquier función $f : \mathbb{R} \to \mathbb{R}$. Uno de los argumentos de la *function* debe ser un el color para la gráfica de la función f.

5. **[3 Pts]**. Tómese una ecuación diferencial del siguiente tipo:

$$\dot{p} = f(p, t) \tag{1}$$

donde $p \in \mathbb{R}^2$ y $t \in \mathbb{R}$. Se puede "resolver" numéricamente con el método de Euler hacia adelante, que consiste en el siguiente método iterativo:

$$\begin{array}{rcl} p_1 & = & p_0 + \delta f(p_0, 0) \\ p_2 & = & p_1 + \delta f(p_1, \delta) \\ p_3 & = & p_2 + \delta f(p_2, 2\delta) \\ & \vdots & \\ p_{n+1} & = & p_n + \delta f(p_n, n\delta) \end{array}$$

donde $p_0 = (x_0, y_0)$ son las condiciones iniciales y $\delta \in \mathbb{R}$ es el "paso de tiempo".
Programa una *function* para resolver numéricamente una ecuación de la forma (1) usando el método de Euler hacia adelante, que regrese la lista de puntos obtenidos ($\{p_0, p_1, \ldots, p_N\}$). A partir de dichos puntos y con segmentos de recta que unan puntos cosecutivos, dibuja las curvas solución para al menos 3 condiciones iniciales distintas.
Sugerencia: Para probar, utilizar la función

$$f(x, y, t) = (y, \mu(1 - x^2)y - x)$$

con algún $\mu \in (0, 4)$, que tiene soluciones interesantes. La ecuación diferencial determinada por esta f es la asociada al oscilador de Van der Pol (por si quieren ver imágenes que muestren que les tiene que salir).
[1 Pt extra]. Dibujar la curva solución con un "gradiente" de color. Por ejemplo de azul a rojo: el primer segmento de recta de p_0 a p_1 en color azul, ir haciendo interpolación de colores en cada segmento, hasta el último segmento p_{N-1} a p_N en color rojo.