# Computational and Differential Geometry
# Homework 3

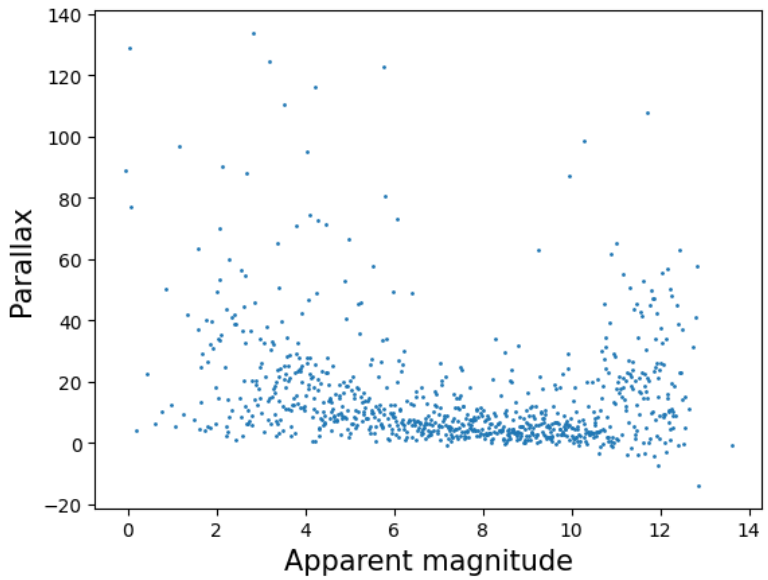
**Professor: Nicolás Avilán Vargas, Ph.D.**

**Indicaciones**

1. Fecha de entrega: 14 de mayo de 2023 hasta las 11:55 pm.
2. Único medio de entrega e-aulas.
3. Formato de entrega: Un único archivo **.ipynb** con códigos en python, descripciones de códigos y procesos, y respuestas a las preguntas (No enviar archivos comprimidos).
4. Solo es permitido el uso de librerías "básicas" (numpy, matplotlib, plotly, seaborn, pandas, sympy, etc). En ningún caso será válida la solución lograda, total o parcialmente, por el uso de una librería especializada para resolver problemas de geometría computacional.
5. La **tarea** debe realizarse **individualmente**.
6. Cualquier tipo de fraude o plagio es causa de anulación directa de la evaluación y correspondiente proceso disciplinario.
7. Las entregas están sujetas a herramientas automatizadas de detección de plagio en códigos.
8. Las tareas no entregadas antes de la hora indicada tendrán calificación de 0.0.

Support each piece of code with a thorough explanation of its methods, techniques, functions, and tricks. Reference your search source (papers, books, tutorials, websites, etc.). Add any necessary bibliographical references or links.

# First part




Read the data in the link star_selection.csv. This file has measurements of stars, these are classified. The variables "Apparent magnitude" and "Paralax" are in the columns **Vmag** and **Plx** in the DataFrame. The figure shows the respective scatterplot.

1. (0.8 points) Implement the 1D Range Searching algorithm to run a query of the interval $[20 : 100]$ for the variable "Paralax". Print the cell IDs for the results of the query.

2. (0.8 points) Implement the 1D Range Searching algorithm to run a query of the interval $[1.8 : 2.5]$ for the variable "Apparent magnitude". Print the cell IDs for the results of the query.

3. (0.9 points) Implement the 2D Range Searching algorithm to run a query of the interval $[20 : 100] \times [1.8 : 2.5]$ for the variables "Paralax" and "Apparent magnitude". Print the cell IDs for the results of the 2D query.

   Do a scatterplot where points in the query have a different color from those which are out of the query.

# Second part

*All the calculations should be developed using the Sympy library in Python.*

4. (1.3 points) Consider the curve

$$\alpha(t) = (a\cos(t), b\sin(t), t)$$

   - Show that torsion $\tau(t)$ for a curve $\alpha(t)$ is given by

$$\tau(t) = -\frac{(\alpha'(t) \times \alpha''(t)) \cdot \alpha'''(t)}{|\alpha'(t) \times \alpha''(t)|^2}$$

     Include the procedure in your notebook using LaTeX.
   - Evaluate the unit vectors $\vec{T}(t)$, $\vec{N}(t)$ and $\vec{B}(t)$ for $\alpha(t)$.
   - Evaluate curvature $\kappa(t)$ and torsion $\tau(t)$ for $\alpha(t)$. Plot the funcitons $\kappa(t)$ and $\tau(t)$, use parameters $a = 2$ and $b = 3$ for plots.
   - Use libraries as matplotlib.pyplot.quiver in order to plot the vectors ($\hat{T}(t)$, $\hat{N}(t)$, $\hat{B}(t)$ ) on the curve.

5. (1.2 points) Consider the toroidal spiral curve

$$\alpha(t) = ((a + \cos(\omega t))\cos(t), (a + \cos(\omega t))\sin(\omega t), \sin(\omega t))$$

   - Evaluate the unit vectors $\vec{T}(t)$ and $\vec{N}(t)$ for $\alpha(t)$, (Try with $\vec{B}(t)$).
   - Evaluate curvature $\kappa(t)$ and torsion $\tau(t)$ for $\alpha(t)$. Use parameters $a = 4$ and $\omega = 10$ for plots.

**Submit**:
Upload to the platform an **.ipynb** file with answers, codes, descriptions and plots.