



SISTEMA DE SEGURIDAD



Proyecto final de Métodos Numéricos
Diego Aarón Moreno Galván

Motivación

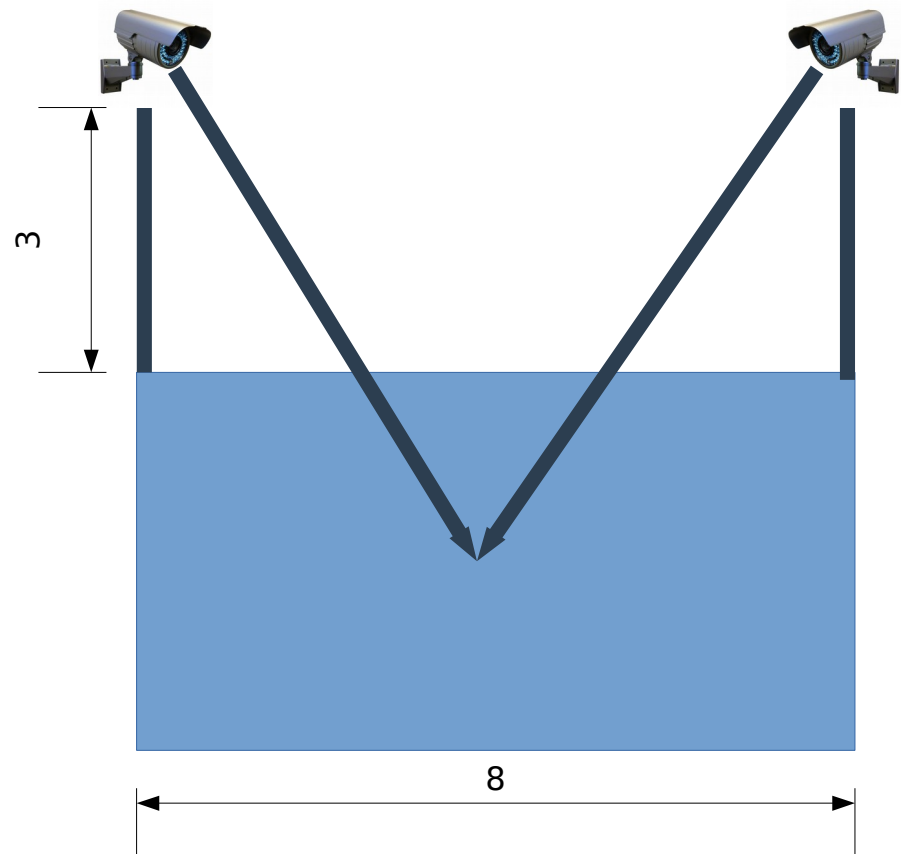


Delimitación del problema

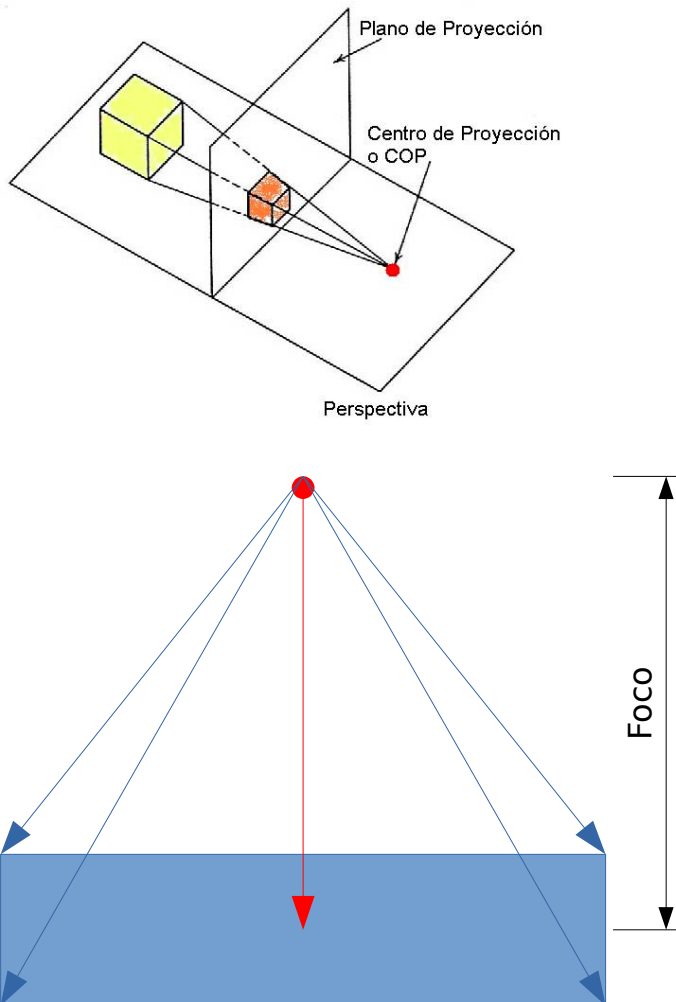
- **Nos enfocaremos solo en lo que implica métodos numéricos.**
- **Generarémos una trayectoria tridimensional.**
- **Proyectarémos en las cámaras.**
- **A partir de las imágenes, reconstruiremos la trayectoria.**
- **Extrapolaremos la posición en la que debemos lanzar el dardo.**

Inicialización del ambiente

- **Patio de tu casa rectangular (este caso 8x8)**
- **Cámaras en dos esquinas (suponemos que están en el origen y en $x=8$).**
- **Las dos a una altura h (este caso 3m).**
- **Apuntan al centro del patio (4,4).**



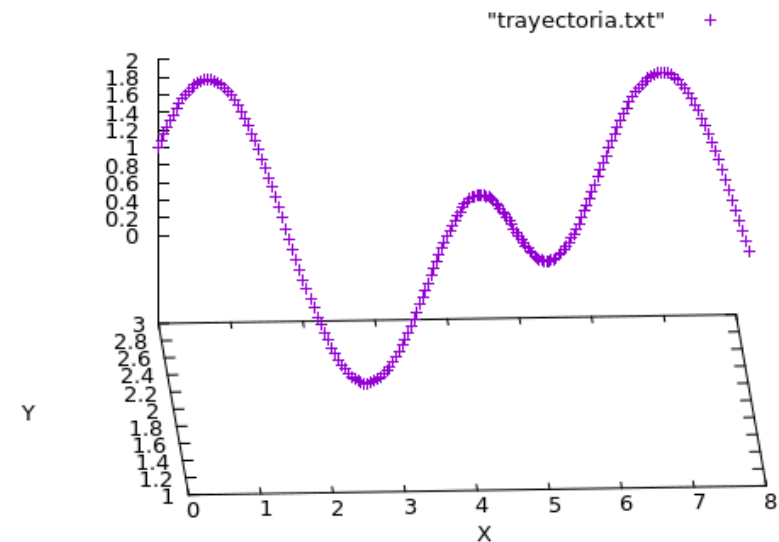
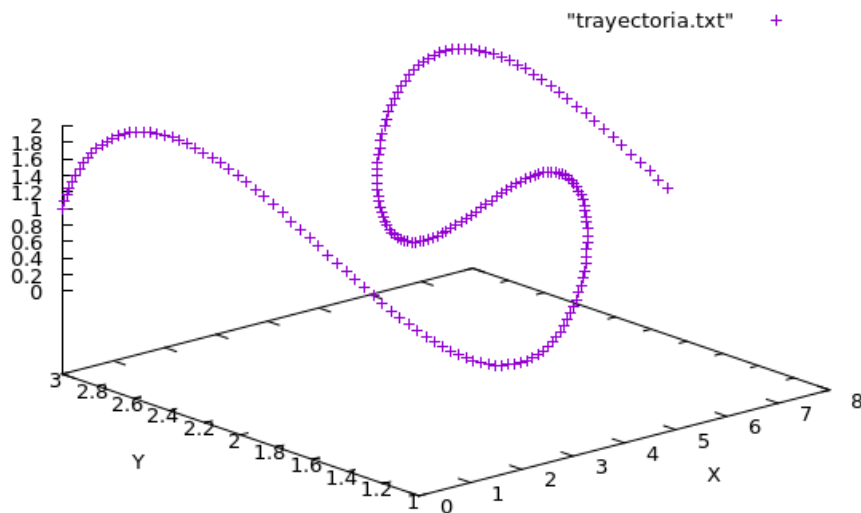
Proyección en cámara



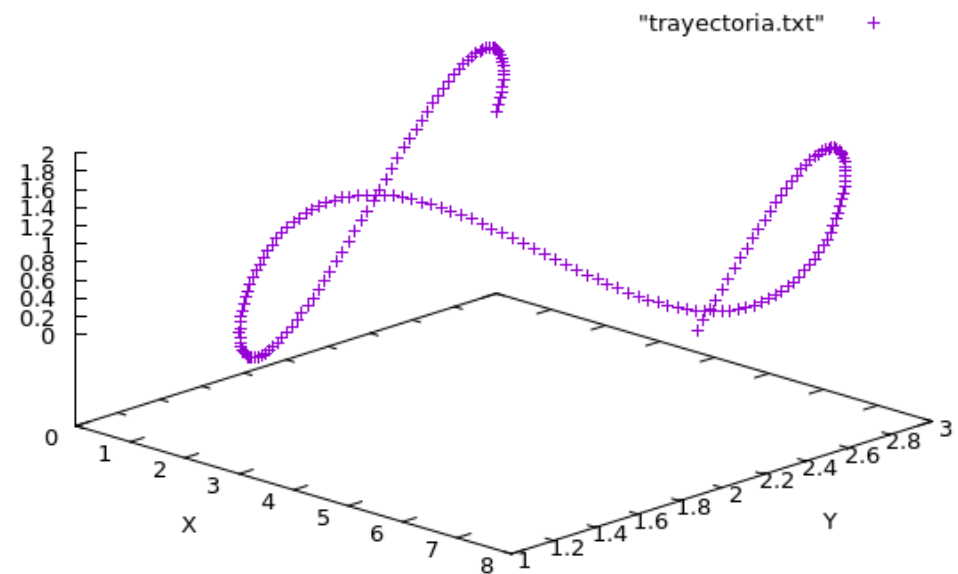
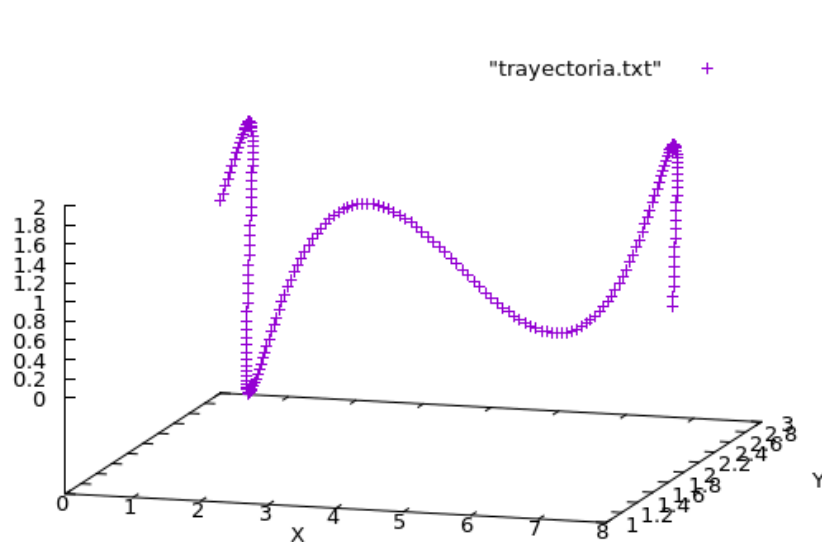
- La cámara de seguridad cuenta con $N \times M$ píxeles (en este caso 800×600 px).
- La distancia del foco al plano proyectivo de la cámara puede variar para mejorar la amplitud de visión.
- Tener cuidado al mejorar la amplitud, pues empeora la resolución.
- Tomar en cuenta la distancia del foco al plano proyectivo.

Generamos la trayectoria

- Para la primera trayectoria usaremos la función:
 $f(t) = (t, \cos(t)+2, \sin(2t)+1)$.
- Normalmente, un ladrón no haría ese trayecto; Solo para poner las cosas más difíciles.

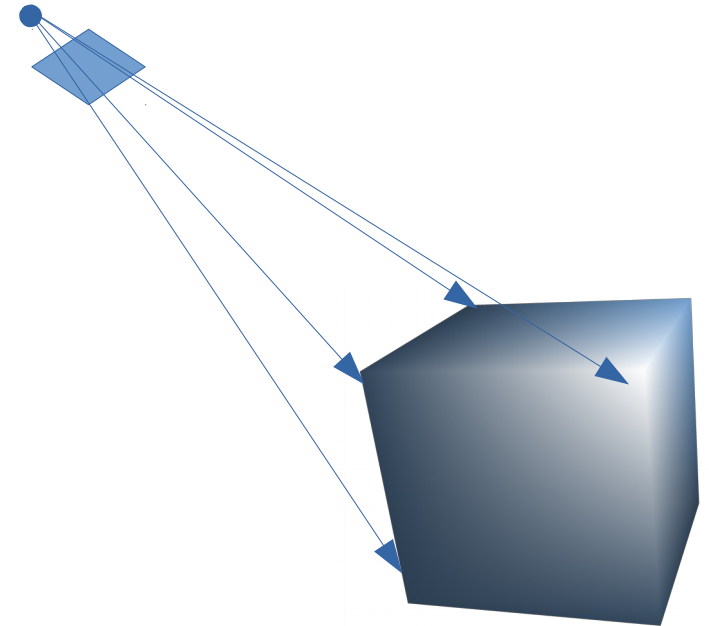


Generamos trayectoria



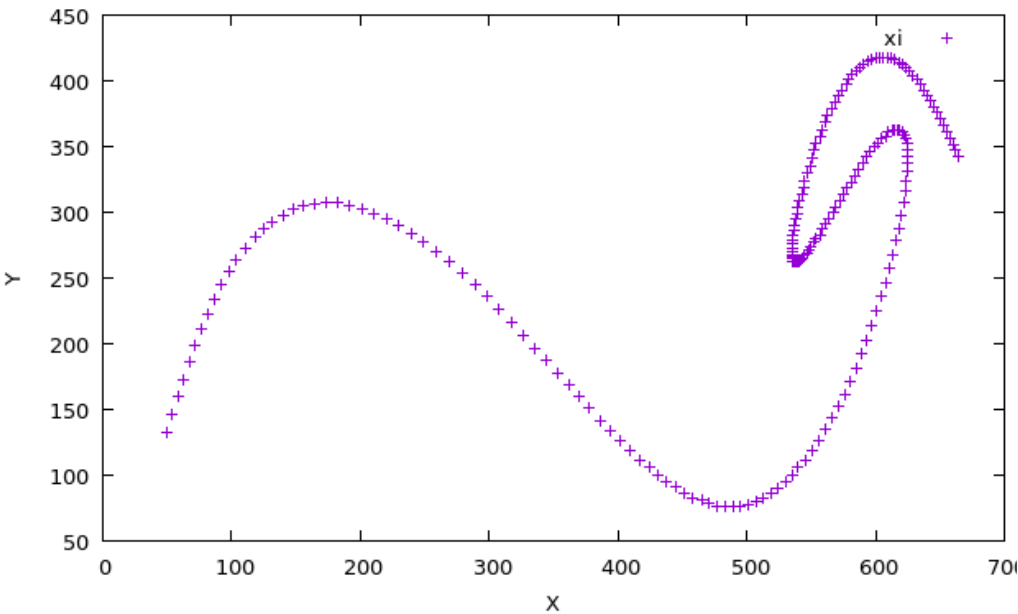
Proyectando a las cámaras

- Para cada cámara, vamos a encontrar la proyección de cada punto en su plano.
- Resolvemos sistemas de ecuaciones, las cuales son ecuaciones del plano proyectivo y la recta que pasa por cada punto y el foco de la cámara.



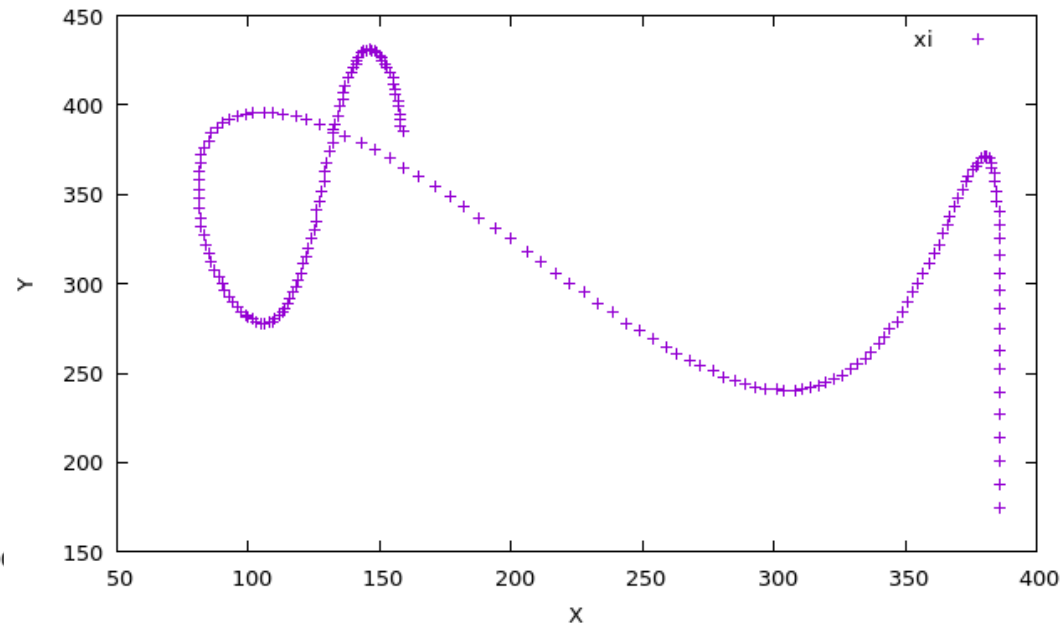
Proyectando a cámaras

Lo que ve la cámara 1



Recordamos que esto está en pixeles.

Lo que ve la cámara 2



Nuestra cámara en este ejemplo es de 800x600 px.



Teniendo las imagenes de cada cámara...
Procedemos a reconstruir la trayectoria

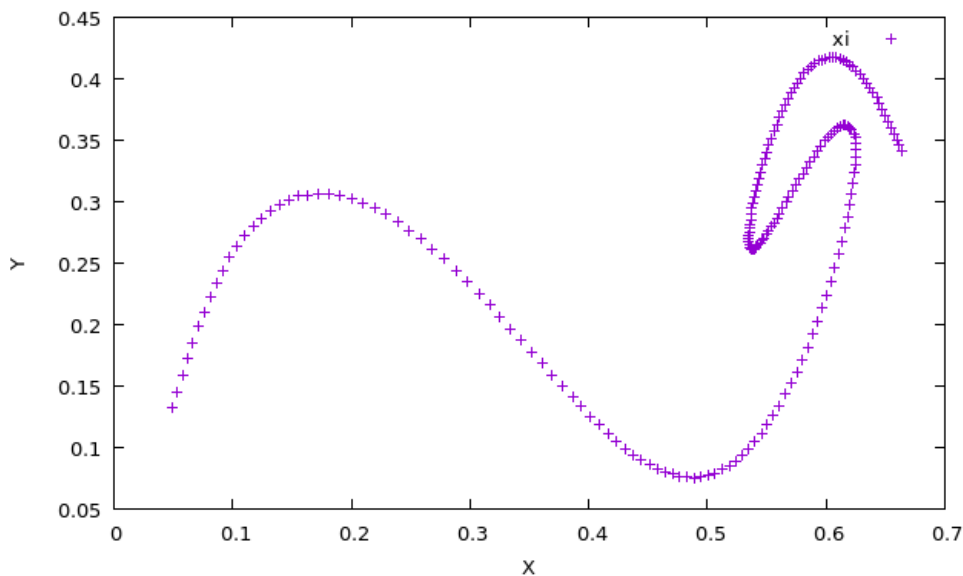
Escalación de imagen

- Supongamos que los pixeles son cuadrados de lado h , (en este ejemplo lo hice con $h=0.001$).
- Tenemos 800 pixeles de ancho, entonces discretizamos de la manera: $x[i]=i*h$ para i de 0 a 800.
- Similarmente, de alto hay 600 pixeles, así la discretización: $y[i]=i*h$ con i de 0 a 600 también funciona.

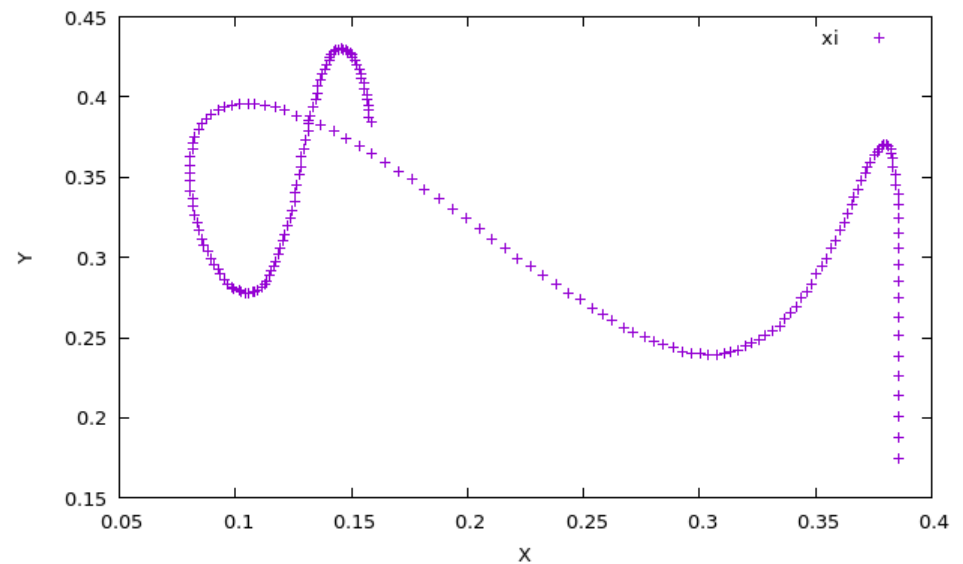
Escalación de imagen

Aquí podemos darnos cuenta que ya vamos a estar arrastrando un error:

- **El error de la resolución de la cámara, pues puede haber varios puntos que fueron proyectados al mismo pixel.**
- **Reducimos todos los puntos en un pixel a solo un punto y perdemos la posición PRECISA de éstos.**



Cámara 1



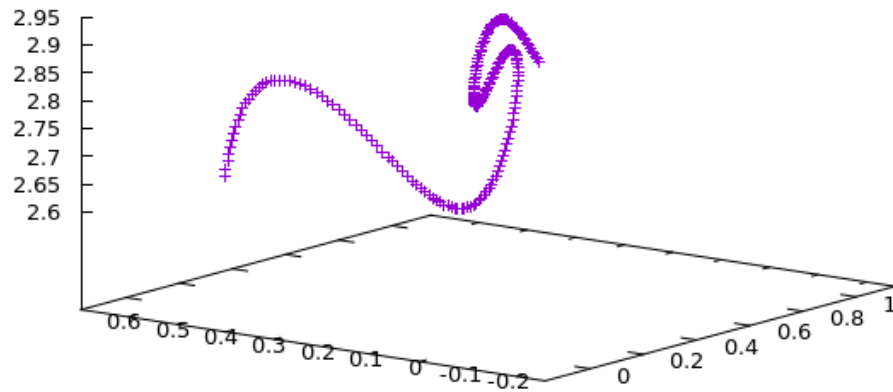
Cámara 2

Trasladamos a plano de cámara

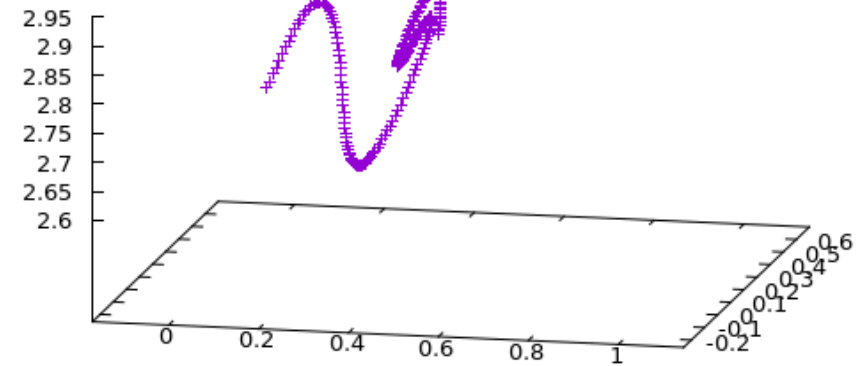
- **Para cada vector de la imagen a escala, lo vamos a rotar en dirección de donde apunta la cámara.**
- **Después lo rotamos verticalmente para que concuerde con el plano de la cámara.**

Trasladamos a plano de cámara

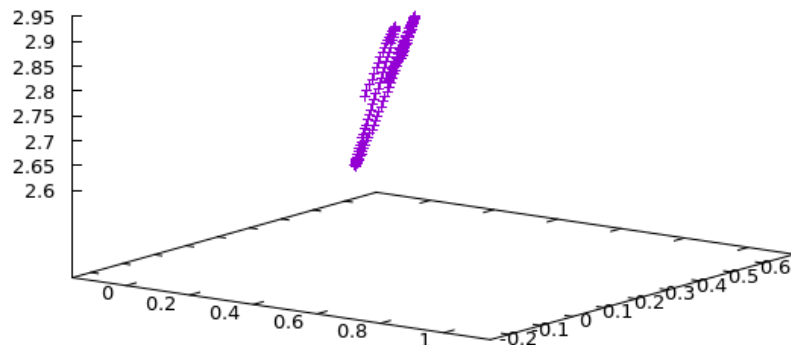
"proyeccion_real_cam1.txt" +



"proyeccion_real_cam1.txt" +



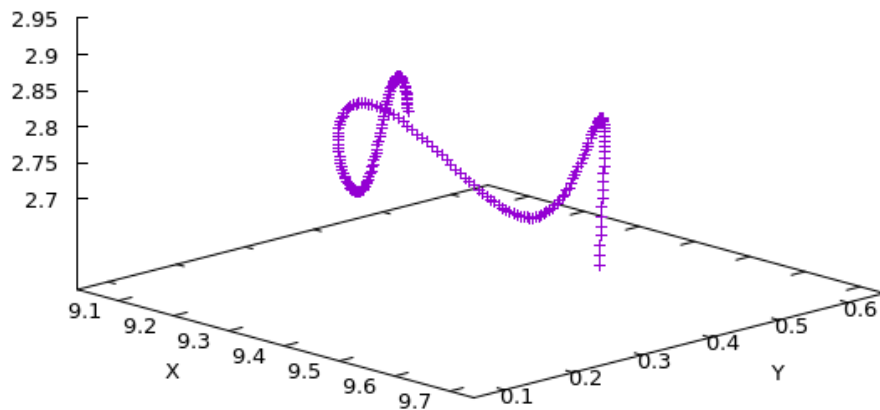
"proyeccion_real_cam1.txt" +



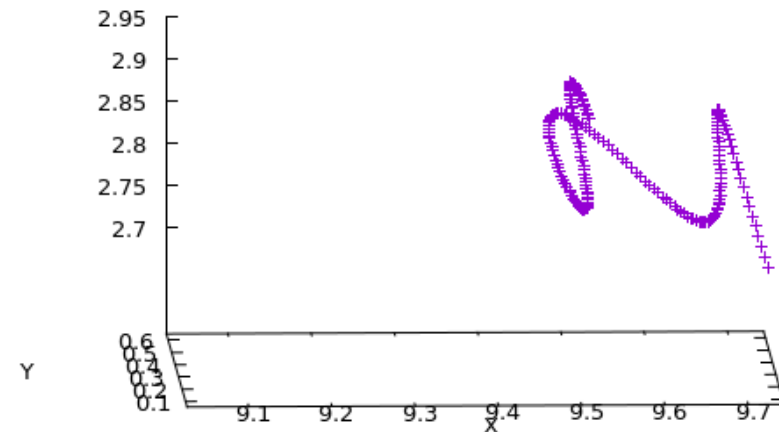
**Traslado de puntos
a plano de
la cámara 1**

Trasladamos a plano de cámara

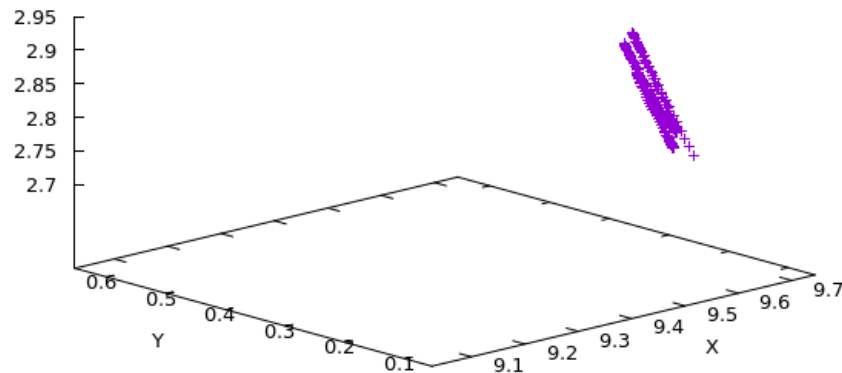
"proyeccion_real_cam2.txt" +



"proyeccion_real_cam2.txt" +



"proyeccion_real_cam2.txt" +



**Traslado de puntos
a plano de
la cámara 2**

Reconstruir trayectoria

Estrategia para reconstruir:

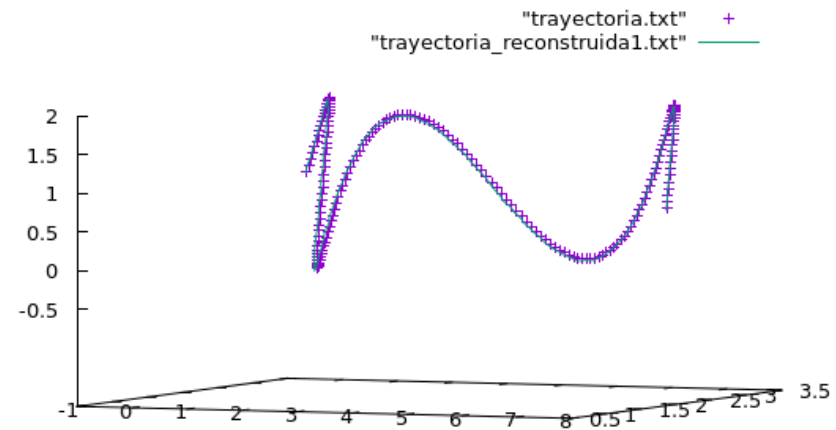
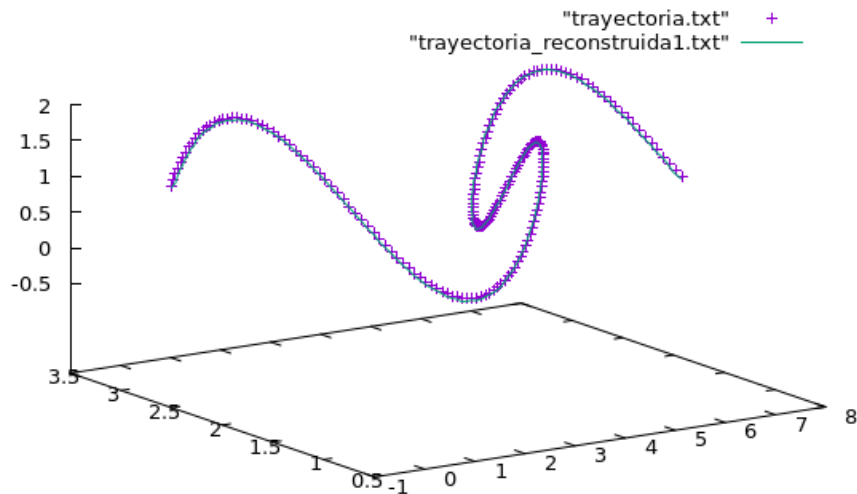
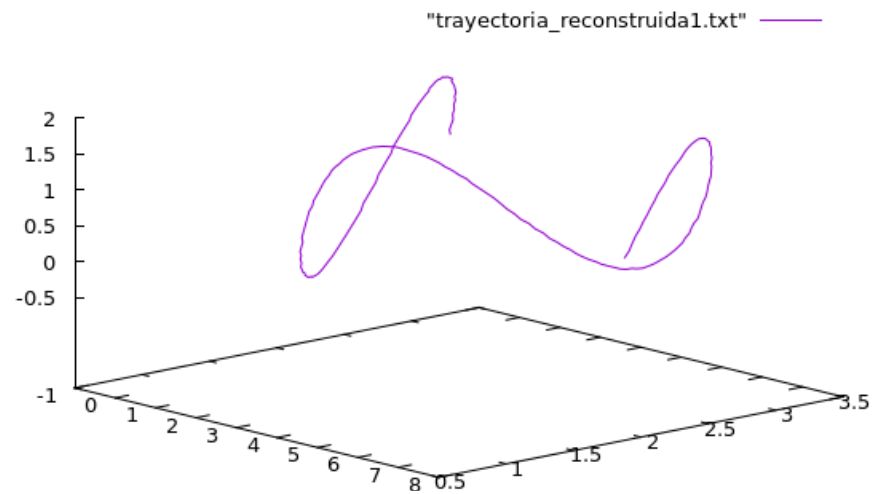
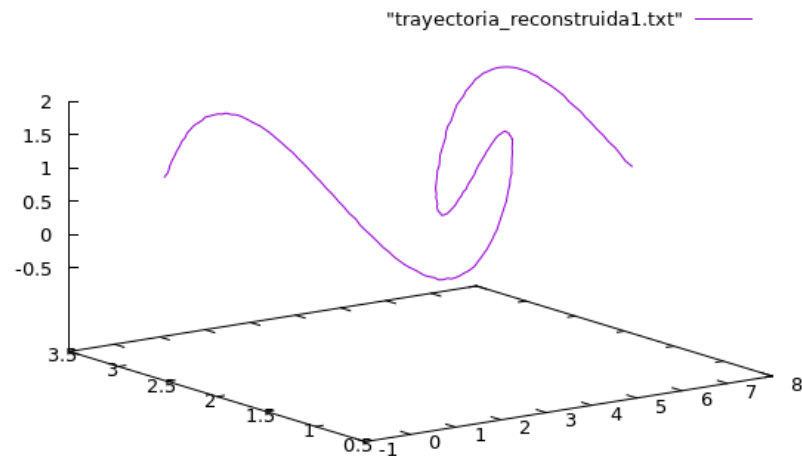
- **Tomar las rectas que pasan por el foco de cada cámara y puntos correspondientes (en tiempo) en los planos de proyección; mirar su intersección.**

Reconstruir trayectoria

Estrategia para reconstruir:

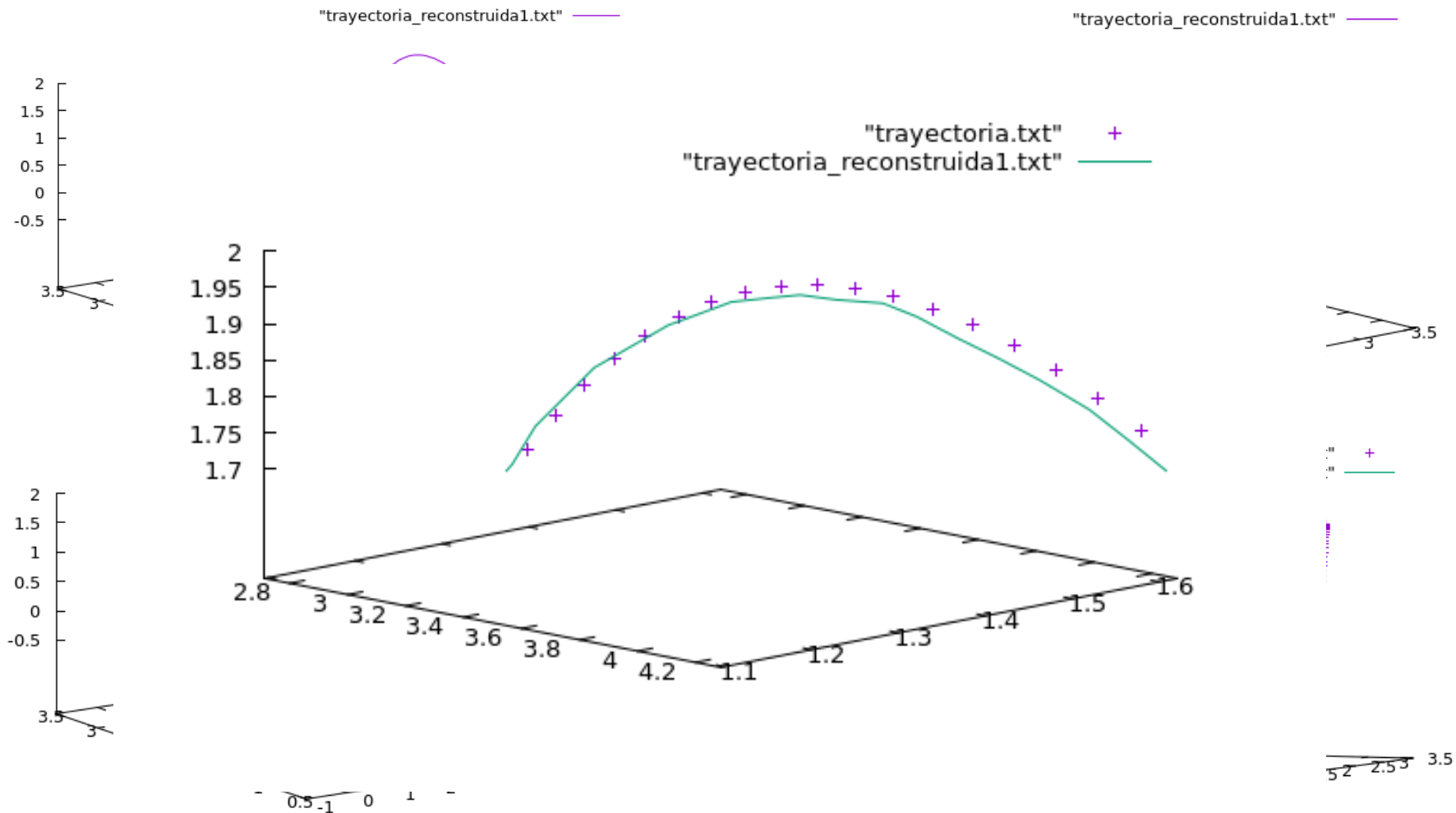
- Tomar las rectas que pasan por el foco de cada cámara y puntos correspondientes (en tiempo) en los planos de proyección; mirar su intersección.
- Tomar en una cámara el plano en dirección de un punto y en la otra, la recta. Así, seguro tiene solución.

Reconstruir trayectoria



Plano en cámara 1 e intersección con recta de cámara 2

Reconstruir trayectoria



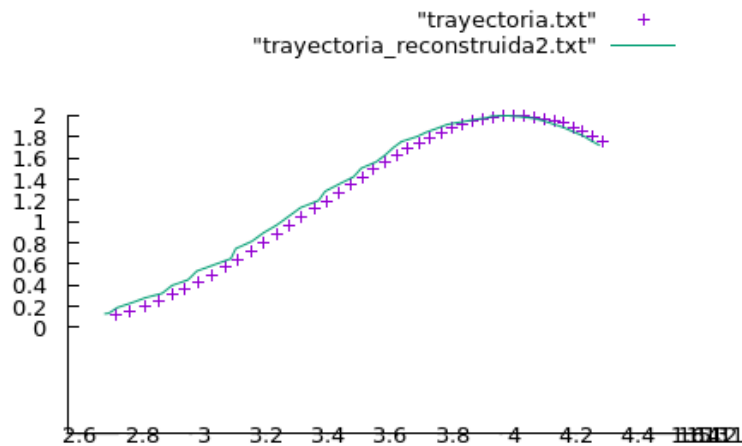
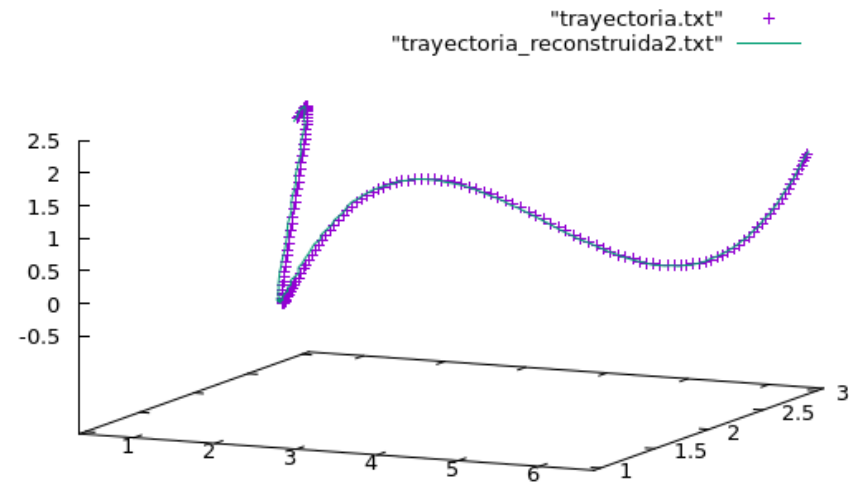
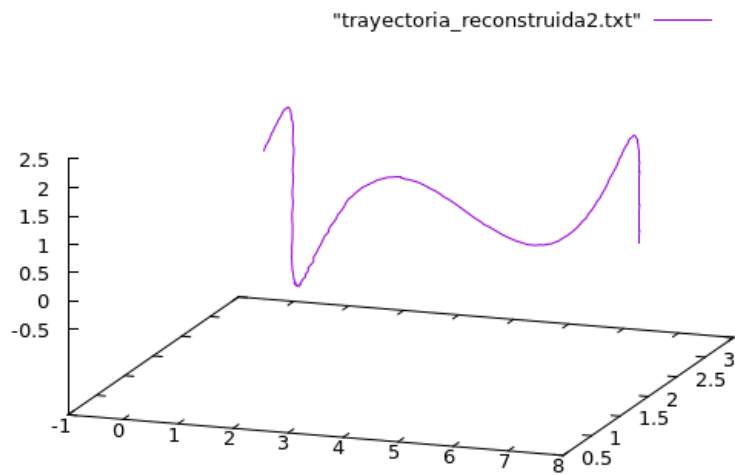
Plano en cámara 1 e intersección con recta de cámara 2

Reconstruir trayectoria

El error de reconstrucción para el caso de generar el plano en la primera cámara y en la otra la recta es de:

```
damorgal@damorgal-hp-envy:~/Documents/Clase Met_Num/Proyecto$ ./pro
Generando el archivo trayectoria.txt ...
Generando el archivo imagen_cam1.txt ...
Generando el archivo imagen_cam2.txt ...
Generando el archivo proyeccion_cam1.txt ...
Generando el archivo proyeccion_cam2.txt ...
Generando el archivo trayectoria_reconstruida1.txt ...
El error de reconstruccion por la izquierda es: 0.507887
Generando el archivo trayectoria_reconstruida2.txt ...
El error de reconstruccion por la derecha es: 0.459012
Generando el archivo trayectoria_reconstruida.txt ...
El error de reconstruccion promedio: 0.446603
Generando el archivo extrapolacion_lineal.txt ...
El error de la extrapolacion lineal es: 0.0125444
Generando el archivo extrapolacion_splines.txt ...
El error de la extrapolacion con splines es: 0.0731615
Generando el archivo proyeccion_real_cam1.txt ...
Generando el archivo proyeccion_real_cam2.txt ...
Hecho
```

Reconstruir trayectoria




Plano en cámara 2 e intersección con recta de cámara 1

Reconstruir trayectoria

El error de reconstrucción para el caso de generar el plano en la segunda cámara y en la otra la recta es de:

```
p  
damorgal@damorgal-hp-envy:~/Documents/Clase Met_Num/Proyecto$ ./pro  
Generando el archivo trayectoria.txt ...  
Generando el archivo imagen_cam1.txt ...  
Generando el archivo imagen_cam2.txt ...  
Generando el archivo proyeccion_cam1.txt ...  
Generando el archivo proyeccion_cam2.txt ...  
Generando el archivo trayectoria_reconstruida1.txt ...  
El error de reconstruccion por la izquierda es: 0.507887  
Generando el archivo trayectoria_reconstruida2.txt ...  
El error de reconstruccion por la derecha es: 0.459012  
Generando el archivo trayectoria_reconstruida.txt ...  
El error de reconstruccion promedio: 0.446603  
Generando el archivo extrapolacion_lineal.txt ...  
El error de la extrapolacion lineal es: 0.0125444  
Generando el archivo extrapolacion_splines.txt ...  
El error de la extrapolacion con splines es: 0.0731615  
Generando el archivo proyeccion_real_cam1.txt ...  
Generando el archivo proyeccion_real_cam2.txt ...  
Hecho
```

Reconstruir trayectoria

- Para minimizar el error, podemos hacer el algoritmo de mínimos cuadrados y obtenemos una nueva trayectoria.
- Con error de:  El cual, mejora las cosas.

```
damorgal@damorgal-hp-envy:~/Documents/Clase Met_Num/Proyecto$ ./pro
Generando el archivo trayectoria.txt ...
Generando el archivo imagen_cam1.txt ...
Generando el archivo imagen_cam2.txt ...
Generando el archivo proyeccion_cam1.txt ...
Generando el archivo proyeccion_cam2.txt ...
Generando el archivo trayectoria_reconstruida1.txt ...
El error de reconstruccion por la izquierda es: 0.507887
Generando el archivo trayectoria_reconstruida2.txt ...
El error de reconstruccion por la derecha es: 0.459012
Generando el archivo trayectoria_reconstruida.txt ...
El error de reconstruccion promedio: 0.446603
Generando el archivo extrapolacion_lineal.txt ...
El error de la extrapolacion lineal es: 0.0125444
Generando el archivo extrapolacion_splines.txt ...
El error de la extrapolacion con splines es: 0.0731615
Generando el archivo proyeccion_real_cam1.txt ...
Generando el archivo proyeccion_real_cam2.txt ...
Hecho
```


Reconstruir trayectoria

- F

e

p

c

r

c

r

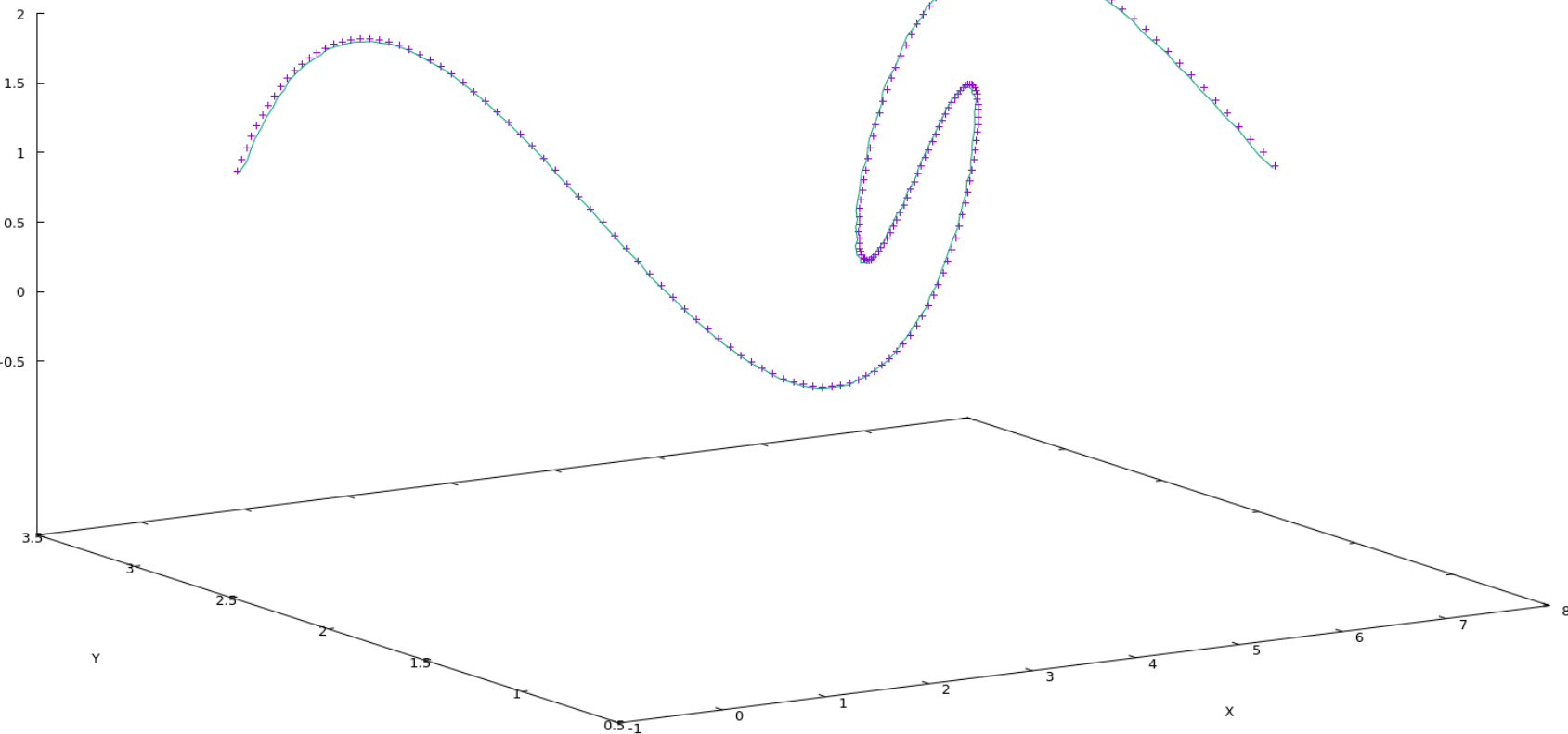
t

- C

E

las cosas.

"trayectoria.txt" +
"trayectoria_reconstruida.txt" —



./pro

Extrapolación de la posición en la que el ladrón va a estar...

Extrapolación lineal

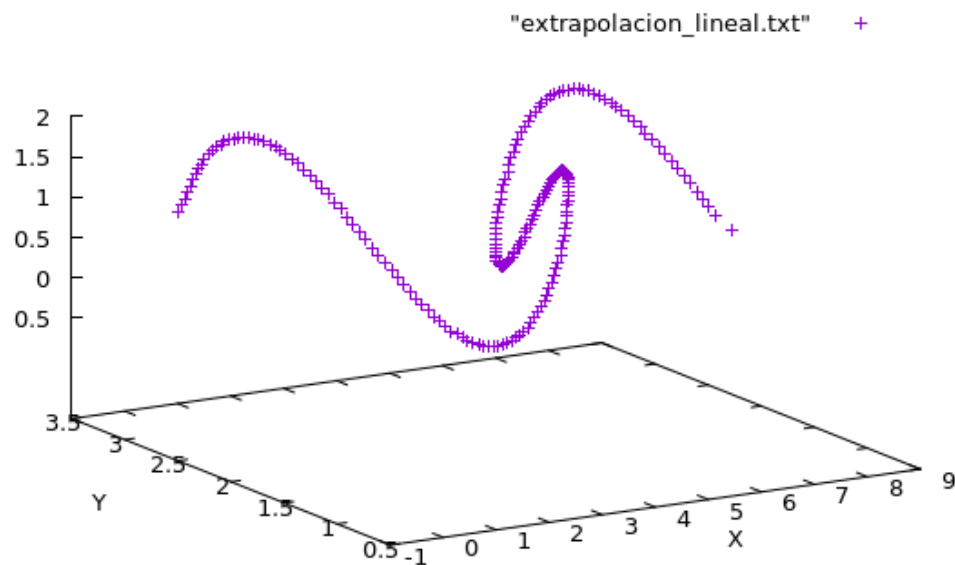
Como estamos considerando intervalos de tiempo muy pequeños y tenemos muchas mediciones, los métodos para extrapolar con polinomios (Lagrange, Newton), no son muy confiables pues son muy inestables.

- Para esto consideraré la opción de extrapolación lineal:**

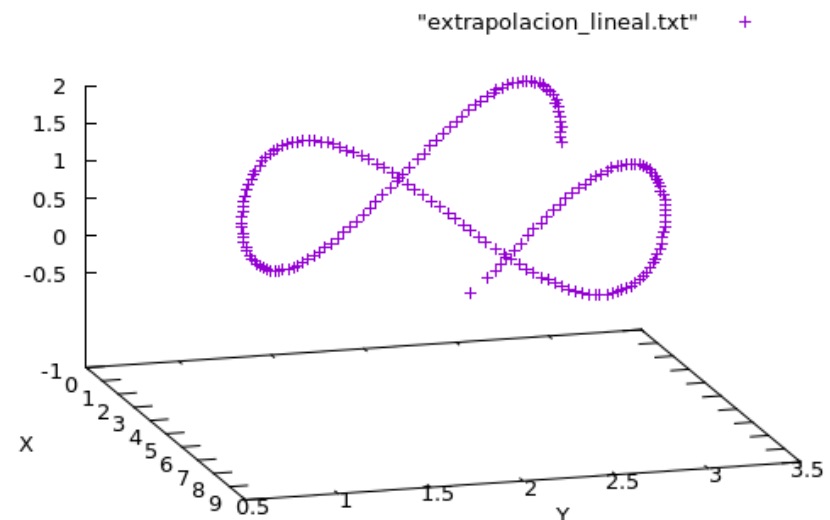
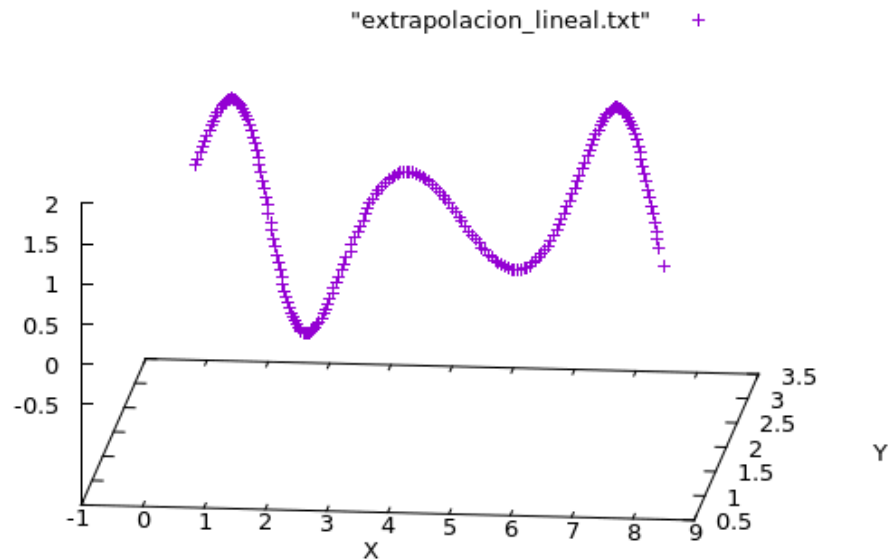
$$y(x^*) = y_{i-1} + (y_i - y_{i-1}) (x^* - x_{i-1}) / (x_i - x_{i-1})$$

Extrapolación lineal

Extrapolamos para medio segundo después...



Y el error de predicción es ...

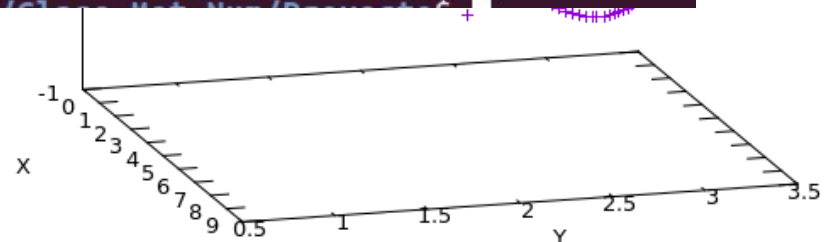


Extrapolación lineal

Extrapolamos para medio segundo después...

```
damorgal@damorgal-hp-envy:~/Documents/Clase Met_Num/Proyecto$ ./pro
Generando el archivo trayectoria.txt ...
Generando el archivo imagen_cam1.txt ...
Generando el archivo imagen_cam2.txt ...
Generando el archivo proyeccion_cam1.txt ...
Generando el archivo proyeccion_cam2.txt ...
Generando el archivo trayectoria_reconstruida1.txt ...
El error de reconstruccion por la izquierda es: 0.507887
Generando el archivo trayectoria_reconstruida2.txt ...
El error de reconstruccion por la derecha es: 0.459012
Generando el archivo trayectoria_reconstruida.txt ...
El error de reconstruccion promedio: 0.446603
Generando el archivo extrapolacion_lineal.txt ...
El error de la extrapolacion lineal es: 0.0125444
Generando el archivo extrapolacion_splines.txt ...
El error de la extrapolacion con splines es: 0.0731615
Generando el archivo proyeccion_real_cam1.txt ...
Generando el archivo proyeccion_real_cam2.txt ...
Hecho
```

Y el error de predicción es ...



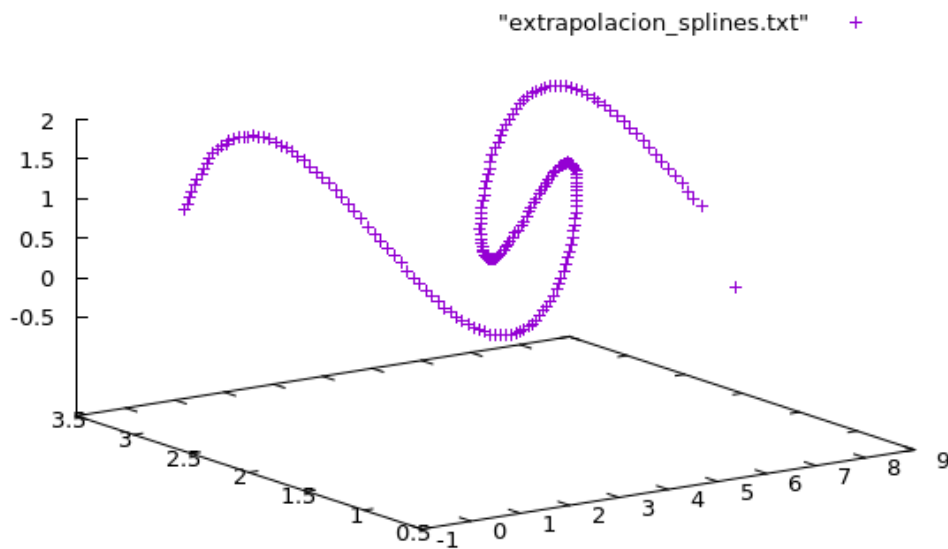
Extrapolación con spline cúbico

Claramente, extrapolación lineal siempre es una línea recta, así que si aumentamos el tiempo a extrapolar, obtendremos error más grande.

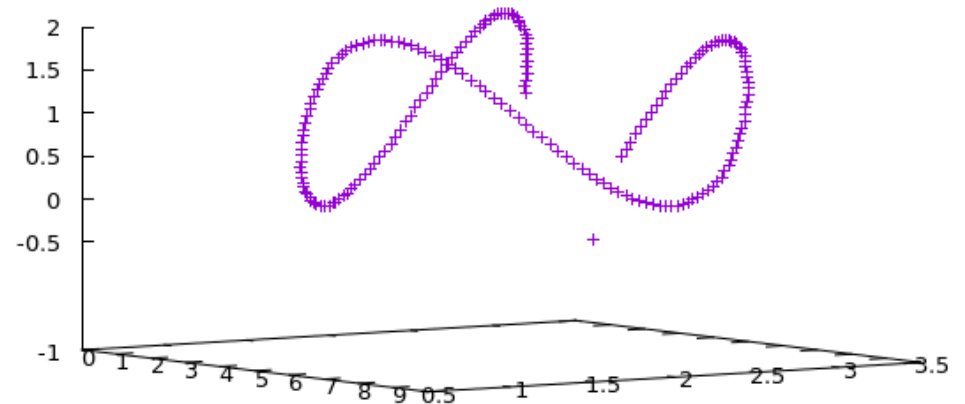
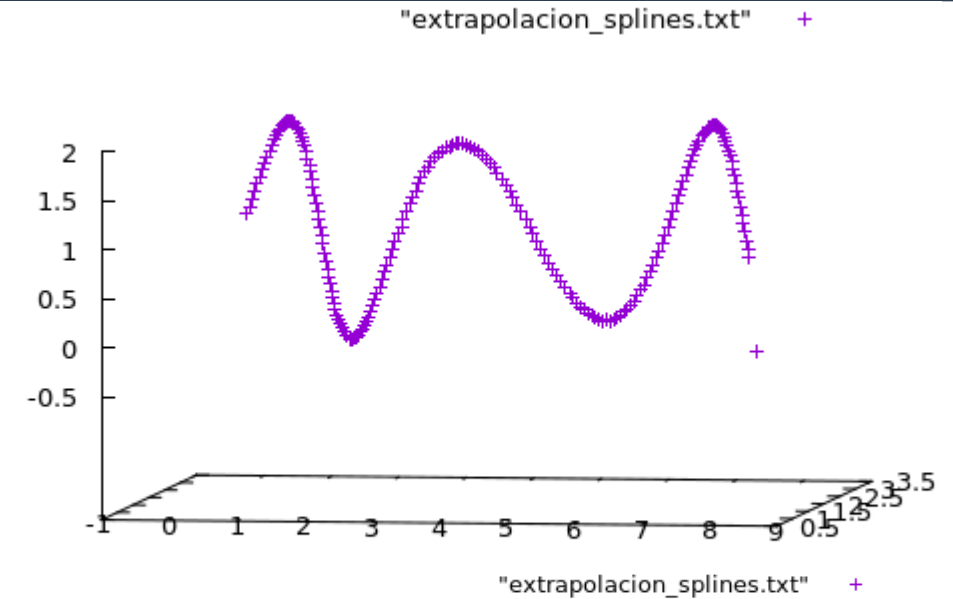
- **Usaremos la técnica de interpolación con splines cúbicos en los puntos interiores**
- **Para extrapolar, consideraré como si el punto estuviera en el último intervalo, es decir, extrapolaremos con el último spline.**

Extrapolación con spline cúbico

Extrapolamos para el doble de tiempo que en lineal ...

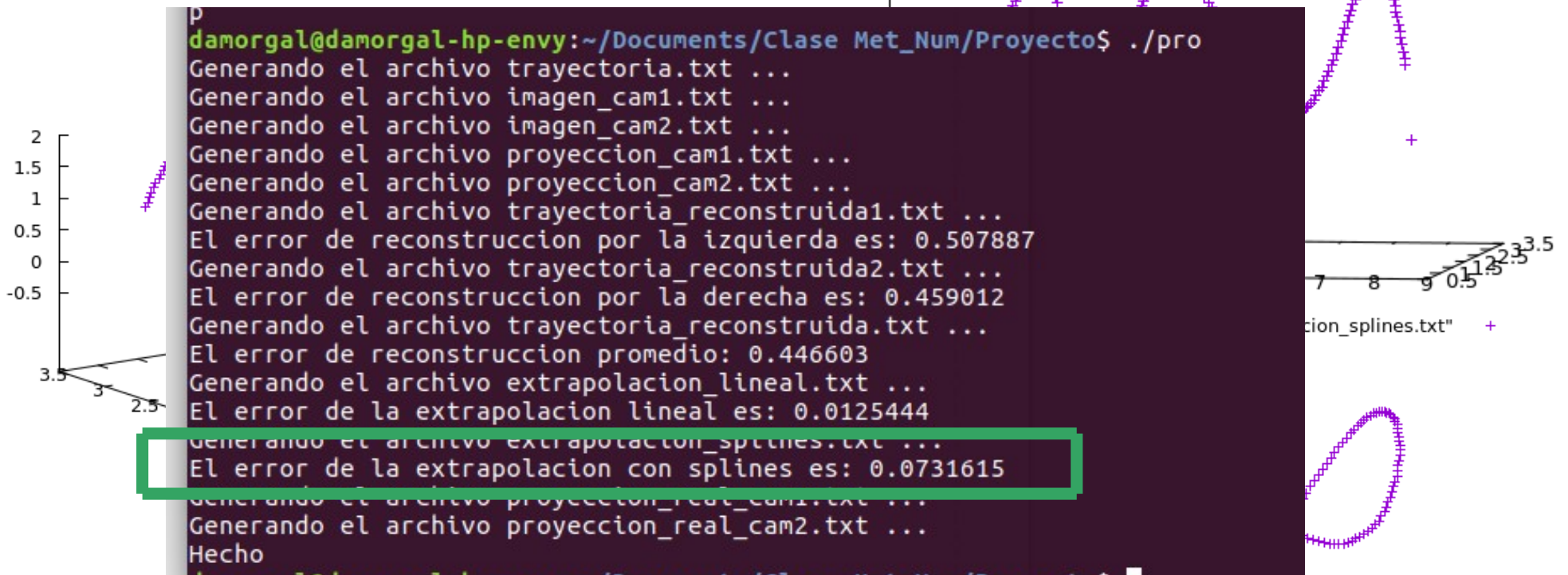


Y obtenemos error de:



Extrapolación con spline cúbico

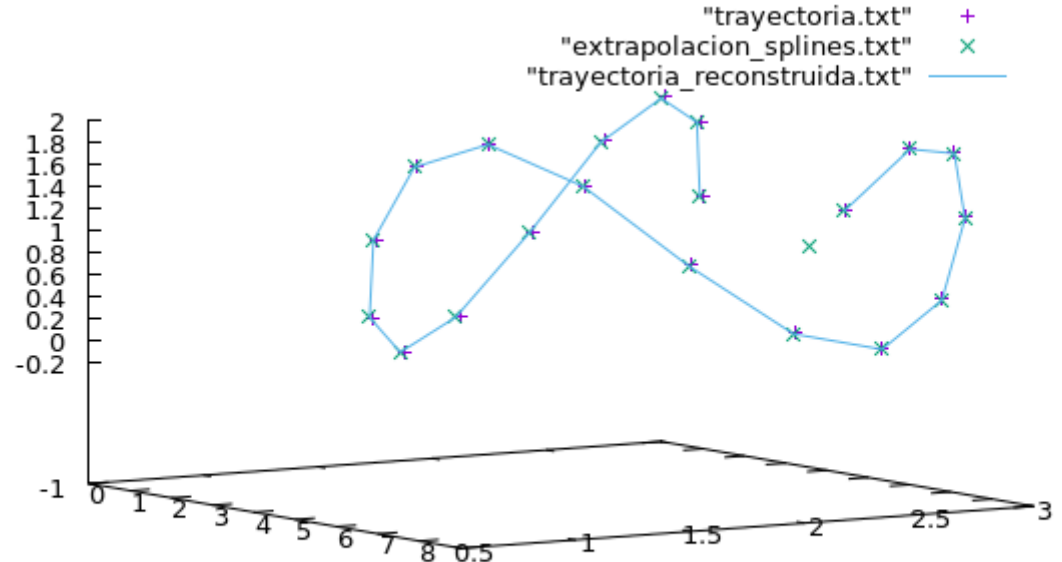
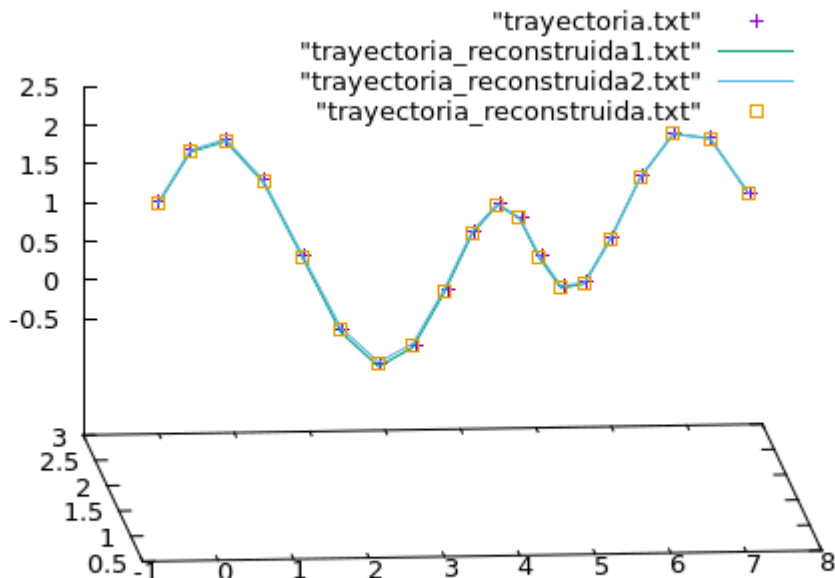
Extrapolamos para el doble de tiempo que en lineal ...



Y obtenemos error de:



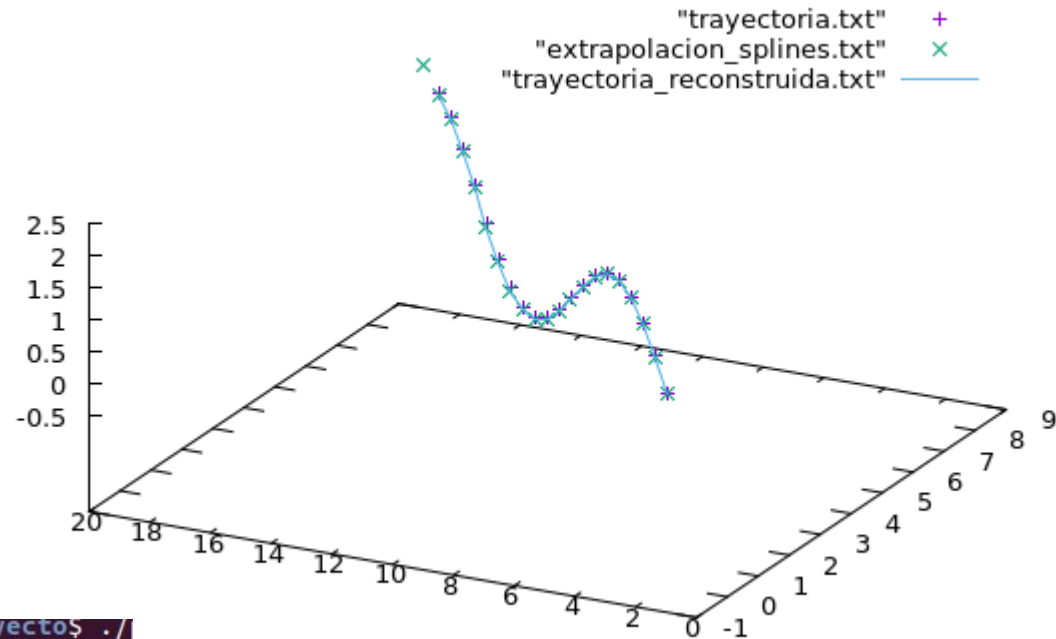
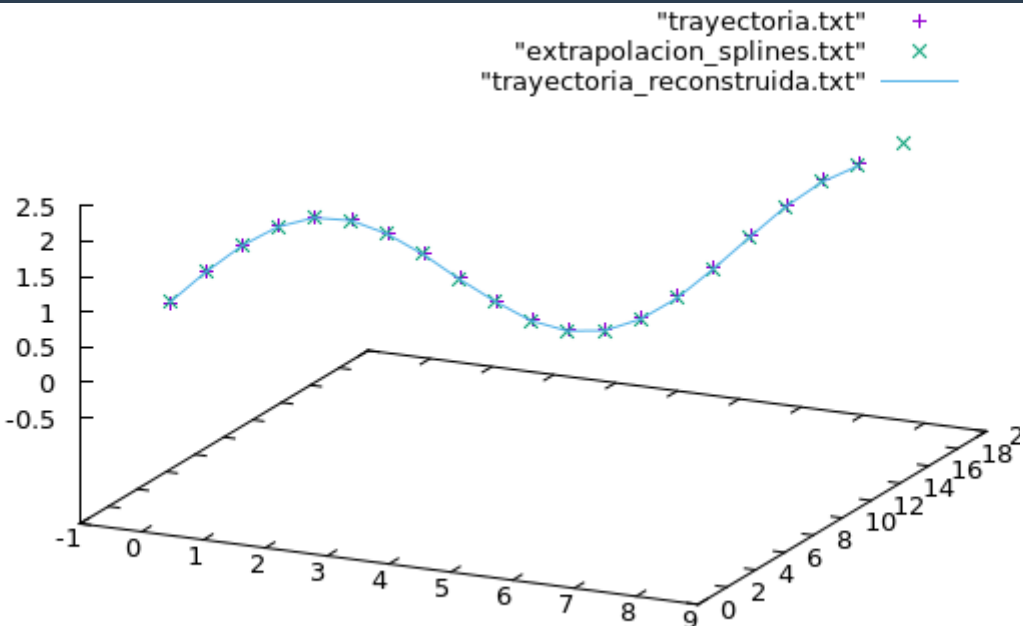
Otros resultados



```
Generando el archivo trayectoria.txt ...
Generando el archivo imagen_cam1.txt ...
Generando el archivo imagen_cam2.txt ...
Generando el archivo proyeccion_cam1.txt ...
Generando el archivo proyeccion_cam2.txt ...
Generando el archivo trayectoria_reconstruida1.txt ...
El error de reconstruccion por la izquierda es: 0.167338
Generando el archivo trayectoria_reconstruida2.txt ...
El error de reconstruccion por la derecha es: 0.151079
Generando el archivo trayectoria_reconstruida.txt ...
El error de reconstruccion promedio: 0.146031
Generando el archivo extrapolacion_lineal.txt ...
El error de la extrapolacion lineal es: 0.0211042
Generando el archivo extrapolacion_splines.txt ...
El error de la extrapolacion con splines es: 0.0471905
Generando el archivo proyeccion_real_cam1.txt ...
Generando el archivo proyeccion_real_cam2.txt ...
Hecho
```

Misma trayectoria;
menos puntos.

Otros resultados



```
damorgal@damorgal-hp-envy:~/Documents/Clase Met_Num/Proyecto$ ./
Generando el archivo trayectoria.txt ...
Generando el archivo imagen_cam1.txt ...
Generando el archivo imagen_cam2.txt ...
Generando el archivo proyeccion_cam1.txt ...
Generando el archivo proyeccion_cam2.txt ...
Generando el archivo trayectoria_reconstruida1.txt ...
El error de reconstruccion por la izquierda es: 0.199582
Generando el archivo trayectoria_reconstruida2.txt ...
El error de reconstruccion por la derecha es: 0.177841
Generando el archivo trayectoria_reconstruida.txt ...
El error de reconstruccion promedio: 0.18508
Generando el archivo extrapolacion_lineal.txt ...
El error de la extrapolacion lineal es: 0.085834
Generando el archivo extrapolacion_splines.txt ...
El error de la extrapolacion con splines es: 0.220819
Generando el archivo proyeccion_real_cam1.txt ...
Generando el archivo proyeccion_real_cam2.txt ...
Hecho
```

$$f(t)=(t, 2t+2, \sin(t)+1)$$