

Instrucciones:

- Se enviará el fichero de cada práctica a la dirección `avalde@ucm.es`.
- El asunto del correo electrónico será: **GC2015 Práctica <poner número aquí>**
- En el cuerpo del mensaje figurarán los nombres, apellidos y correos electrónicos de los alumnos que envían la práctica. Salvo casos muy excepcionales y bien justificados, el grupo de alumnos se mantendrá constante a lo largo de todo el curso.
- El plagio implicará que los alumnos pasarán a ser evaluados de la asignatura mediante la realización del examen final de la misma.
- Si la práctica contiene preguntas, deben responderse en un fichero aparte (PDF, escaneo de manuscrito, etc. No se admiten ficheros word).
- Cada práctica deberá enviarse antes del día y hora indicado como límite. No se aceptarán envíos pasado ese momento.

P4: Se trata de implementar el algoritmo de intersección de curvas de Bézier visto en clase. La práctica se entregará en un fichero con el nombre `intersection_bezier.py` escrito en python 2.

- En el fichero `intersection_bezier.py` se definirá una clase de nombre `IntersectionBezier` que será la que realice los cálculos de las intersecciones.
- En la documentación al principio del módulo figurarán los nombres de los autores del código.
- Dicha clase tendrá un método `__call__(self, P, Q, epsilon)` que recibirá los polígonos de control de las dos curvas en forma de arrays de numpy, con `P.shape = (n, 2)` y `Q.shape = (m, 2)` y un umbral, `epsilon > 0` que controlará si se aproxima el polígono de control mediante un segmento. Devolverá los `K` puntos de intersección calculados como un array de numpy de dimensiones `(K, 2)`.
- Habrá también un método `__plot__(self, k=3)` que producirá el dibujo de las curvas y las intersecciones.
- El dibujo de las curvas consistirá en mostrar los polígonos compuestos mediante `k` subdivisiones sucesivas, de acuerdo con el algoritmo visto en clase.
- Como criterios de evaluación se usarán las unidades de prueba que pasará `__call__`, la velocidad de ejecución de `__call__` y la elegancia y claridad del código.

- Opcional: hacer que la práctica sea interactiva, de forma que ejecutando
`>>> python intersection_bezier.py`
se abra una ventana gráfica en la que introducir los polígonos y en la que se pueda modificar `k` y `epsilon` usando el teclado.

Límite para entregar esta práctica: lunes 6 de abril, 23:59h.