Actividad grupal: Polinomios de Chebyshev y visualizaciones en Jupyter Notebooks

Objetivos

En esta actividad se llevarán a cabo procedimientos de ajuste e interpolación en Python empleando las bibliotecas NumPy y los módulos interpolate y optimize de SciPy.

El ajuste de curvas o aproximación resulta imprescindible para adecuar los datos obtenidos a partir de mediciones en conformidad con un modelo predefinido.

La interpolación se emplea para construir una curva que atraviese un conjunto de puntos, logrando un ajuste perfecto. En situaciones en las que la cantidad de puntos sea reducida, es posible utilizar polinomios interpoladores; no obstante, si la cantidad de puntos es mayor, se requerirá emplear trazadores (conocidos como *splines* en inglés).

Un importante número de conclusiones se podrán extraer de las visualizaciones generadas a partir de los datos y de los polinomios o trazadores utilizados.

Descripción de la actividad

Para llevar a cabo la actividad es posible utilizar las funciones proporcionadas por las bibliotecas mencionadas. En caso de que no se encuentre disponible algún método específico, se deberá realizar su implementación.

Tarea 1. Interpolación polinómica

La interpolación polinómica es ampliamente utilizada como método para obtener curvas que atraviesen un conjunto de puntos. Sin embargo, surge un desafío cuando el número de puntos aumenta significativamente. En tales casos, se presenta lo que se denomina como el fenómeno de Runge, en el cual el polinomio oscila de manera considerable.

Para entender este efecto y ver una representación gráfica pueden consultar el [artículo](https://es.wikipedia.org/wiki/Fen%C3%B3meno_de_Runge) sobre el fenómeno de Runge en Wikipedia.

Por lo tanto, la elección de los nodos de interpolación, es decir, los valores x de los puntos, se convierte en una tarea crucial, ya que el error de ajuste variará en función de esto.

Durante la actividad se trabajará con tres funciones diferentes y se realizará una comparación entre tres métodos de interpolación polinómica, así como la interpolación con *splines* utilizando nodos equidistantes o nodos obtenidos a partir de los polinomios de Chebyshev.

Profundicen sobre la interpolación con la [presentación](https://personales.unican.es/segurajj/interp_p.pdf) de Javier Segura.

Funciones

Las funciones son:

* .

Consideraciones

* Número de nodos:
* 11 nodos: realizar los supuestos considerando únicamente 11 nodos.
* 21 nodos: realizar los supuestos considerando únicamente 21 nodos.
* Obtención de los nodos:
* Nodos equiespaciados. Los nodos se distribuyen de manera uniforme en el intervalo.
* Raíces del polinomio de Chebyshev. Los polinomios de Chebyshev son una familia de polinomios ortogonales. Para obtener los nodos se calcularán los polinomios hasta el grado correspondiente al número de puntos o nodos de interpolación que se desean considerar. Las raíces de dicho polinomio serán los puntos de interpolación.
* Para esto, se puede usar el módulo chebyshev de la biblioteca NumPy. Se deberá obtener el polinomio del grado correspondiente y sus raíces para obtener los nodos de interpolación.

Para obtener información sobre los polinomios de Chebyshev puede consultar este [enlace](http://www.sc.ehu.es/sbweb/fisica3/especial/chebyshev/chebyshev.html).

* Métodos de interpolación:
* Método de interpolación baricéntrica, mediante la función barycentric\_interpolate del módulo interpolated de scypy.

Esta función requiere la lista de las x e y de los puntos de interpolación y un *array* en el que evaluar el polinomio de interpolación.

* Método de interpolación de Lagrange mediante la función lagrange del módulo interpolated de ScyPy.
* Método de diferencias dividas de Newton. Se debe implementar en Python. La implementación debe ser explicada y justificada.

En este caso se podría usar la biblioteca sympy para realizar cálculos simbólicos, que permite la definición de expresiones algebraicas, es decir, el polinomio de interpolación con la función lambdify. En este método se debe calcular una tabla que se construirá como se muestra en el ejemplo.

Dados tres puntos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| i |  |  |  |  |
| 0 | 0 | -1 |  |  |
| 1 | 1 | -1 |  |  |
| 2 | 2 | 7 |  |  |

El polinomio interpolador es

Puede consultar más información en

<https://personales.unican.es/segurajj/tema3-2.pdf>

* Cálculo de los errores de los resultados obtenidos donde:

Siendo , la función que se va a aproximar y el polinomio obtenido.

Para el cálculo del error se pueden usar funciones de los módulos de algebra lineal de NumPy o SciPy.

* Cálculo de los tiempos de obtención del polinomio por cada método. Puede usar time de la biblioteca time para medir el tiempo de ejecución de cada método.
* Para la obtención de las gráficas debe tener en cuanto los valores de las x para los que se hace la representación.

Se pide que para cada una de las funciones:

* Obtengan los puntos de la función en los nodos (11 y 21) de interpolación seleccionados. Expliquen cómo los construye para cada una de las funciones.
* Usen esos puntos para obtener el polinomio de interpolación según el método elegido. Expliquen los resultados obtenidos (polinomio) para cada función, método y tipo y número de nodos.
* Calculen el error de interpolación en cada caso.
* Calculen los tiempos de cálculo de los polinomios.
* Dibujen las gráficas de la función real y del polinomio obtenido. Pueden usar una misma figura para cada método o pueden incorporarlas todas en la misma. Indiquen las conclusiones que se obtienen del análisis de las gráficas.
* Muestren las conclusiones obtenidas indicando las diferencias encontradas, contrastando los valores obtenidos en los errores, en el tiempo de cálculo y el análisis de las gráficas.

Tarea 2. Interpolación con *splines*

Esta interpolación utiliza *splines*, que es una curva diferenciable definida en trozos mediante polinomios.

La principal ventaja de este método estriba en que se elimina el fenómeno de Runge o, lo que es lo mismo, se eliminan las oscilaciones, ya que los polinomios que se utilizan son de un grado bajo.

A continuación se deben realizar las siguientes tareas, siguiendo los mismos pasos que en la tarea anterior:

* Obtengan los puntos de la función en los nodos de interpolación seleccionados. Expliquen cómo los construye para cada una de las funciones.
* Usen esos puntos para obtener el polinomio de interpolación con *splines* cúbicos. Expliquen el resultado obtenido para cada función, tipo y número de nodos.
* Calculen el error de interpolación en cada caso.
* Calculen los tiempos de cálculo de los polinomios.
* Dibujen las gráficas de la función real y del polinomio obtenido.
* Presenten las conclusiones obtenidas indicando las diferencias encontradas, contrastando los valores obtenidos en los errores, en el tiempo de cálculo y el análisis de las gráficas.

Rúbrica

|  |  |  |  |
| --- | --- | --- | --- |
| Polinomios de Chebyshev y procesamiento… | Descripción | Puntuación máxima  (10) | Peso  100% |
| Nodos de interpolación | * Se definen los nodos (11 y 21 nodos). * Se obtienen los valores de las funciones en esos nodos. * Se explica el procedimiento. | 2 | 20% |
| Cálculo de los polinomios de interpolación | * Método de interpolación baricéntrica. * Método de interpolación de Lagrange. * Método de diferencias dividas. * Interpolación por *splines.* * Se explica la aplicación de cada uno de los métodos. * Se muestra el polinomio de interpolación. | 3  (0,5 puntos cada método y 1 para diferencias divididas, resto explicación y visualización polinomio). | 30% |
| Error de los métodos y cálculo de los tiempos | * Se calcula el error absoluto en cada caso. * Se calcula el tiempo de cálculo del polinomio en cada caso. | 2 | 20% |
| Gráficas de la función y de los polinomios | * Se dibujan las gráficas de las funciones y sus polinomios interpoladores. * Se analizan las gráficas. | 2 | 20% |
| Conclusiones | * Se extraen las conclusiones a la vista de los datos de error, tiempo y análisis de las gráficas. | 1 | 10% |
|  |  | **10** | **100 %** |

Forma de entrega

Este trabajo deberá entregarse en formato editable (.doc, .docx, .odf, .rtf) o un *notebook*. Se debe adjuntar el código de las implementaciones si no se entrega con un *notebook*.

Extensión máxima de la actividad

Diez páginas con fuente Georgia 11 e interlineado 1,5.