

ILI-286: Computación Científica II  
Tarea #1  
*“Computación numérica de valores propios”*

SCT II-2016

12 de Agosto de 2016

## Descripción y un poco de teoría

En la siguiente tarea, usted realizará un análisis de complejidad algorítmica y de convergencia sobre los métodos para la obtención de valores propios estudiados en el curso. Para esto, comparará de acuerdo a distintos parámetros los algoritmos Power Iteration, Inverse Power Iteration, Rayleigh Quotient Iteration y Unshifted QR Algorithm. Adicionalmente, su creatividad y conocimientos serán puestos a prueba con el desafío de crear un algoritmo para resolver un determinado problema.

Para la realización de esta experiencia necesitará crear matrices simétricas y definidas positivas de distintos tamaños conociendo de antemano sus valores propios, para así poder comparar con sus resultados. Con estos cuatro simples pasos usted podrá lograr esto:

- Cree una matriz cuadrada ( $B$ ) de números aleatorios que posea el tamaño deseado para la matriz final.
- Realice la descomposición QR de la matriz  $B$ , es decir,  $B = QR$ .
- Cree una matriz diagonal ( $\Lambda$ ) en donde se encontrarán los valores propios que deberá tener la matriz final. Para la elección de valores propios, utilice la siguiente fórmula:

$$\begin{aligned}\lambda_1 &= \frac{1}{2} \\ \lambda_2 &= \frac{1}{3} \\ &\vdots \\ \lambda_i &= \frac{1}{1+i} \\ &\vdots \\ \lambda_n &= \frac{1}{1+n}\end{aligned}$$

- Finalmente cree la matriz  $A = Q\Lambda Q^*$ , que es una matriz simétrica, positiva definida y cuyos valores propios han sido especificados por usted mismo.

Considere utilizar la misma semilla para la creación de las matrices aleatorias y hacer que sus experimentos sean reproducibles. Para esto debe utilizar la rutina `numpy.random.seed`.

## Desarrollo

1. Implemente los algoritmos Power Iteration, Inverse Power Iteration, Rayleigh Quotient Iteration y Unshifted QR Algorithm. En caso de ocupar los algoritmos presentados en los *notebooks* del curso, modifíquelos como estime conveniente para mejorar su desempeño sin cambiar la estructura principal del método. Adicionalmente, implemente el algoritmo que fue explicado en la descripción de la tarea para poder crear matrices simétricas con valores propios conocidos.
2. Estime la complejidad algorítmica de cada método implementado. Para esto, cree matrices  $A_{n \times n}$  para  $n = 10, 100, 1000, 2000$  y tome el tiempo que demora cada algoritmo en ejecutar una iteración. Para lograr una mayor precisión en su medición, realice un gran número de iteraciones y divida el tiempo resultante por dicho número. Adicionalmente, se recomienda que repitan varias veces el experimento para cada algoritmo y calculen el promedio de los tiempos observados. Muestre los resultados obtenidos en un único gráfico y comente al respecto.
3. Analice la convergencia de cada algoritmo calculando el error  $(|\lambda - \lambda_1^{(k)}|)$  obtenido hasta que sea menor o igual a  $1e-10$ , donde  $\lambda_1$  es el valor propio dominante y  $k$  es el número de iteración. En esta ocasión, debe graficar el error respecto al número de iteraciones. Se tendrán cuatro gráficos, uno por cada matriz  $A_{n \times n}$  con  $n = 10, 100, 1000, 2000$ , y cada uno mostrando el desempeño de los cuatro métodos a la vez. Nuevamente, compare y concluya sobre lo observado.
4. A continuación, se le pide que combine las observaciones producto de las preguntas anteriores, esto graficando el error obtenido (eje  $x$ ) en relación al tiempo de ejecución de cada algoritmo (eje  $y$ ), multiplicando el número de iteraciones ejecutadas al momento del cálculo del error (Pregunta 1) por el tiempo que demora cada iteración (Pregunta 2). Nuevamente tendrá cuatro gráficos, uno por cada tamaño de la matriz, en cada cual se mostrará el desempeño de los cuatro métodos. Haga un análisis en conjunto sobre los distintos parámetros estudiados, respondiendo a preguntas como:
  - Si se necesita realizar el cálculo de valores propios en un determinado tiempo acotado, ¿cuál es el error que puede esperarse de un determinado método? ¿Cuál es el método que elegiría en este caso?
  - Si se desea obtener un error no más allá de una determinada tolerancia, ¿cuánto tiempo se tardará la ejecución de un determinado algoritmo?

Estas preguntas son a modo de ejemplo y referencia, usted debe definir como plantear sus observaciones y comparaciones. Se espera que sus conclusiones sean completas a la hora de analizar los datos de forma agrupada.

## Instrucciones:

- (a) La estructura del laboratorio es la siguiente:
  - (a) Título, nombre de estudiante, email y rol.
  - (b) Introducción.
  - (c) Desarrollo y análisis de resultados.
  - (d) Conclusiones.
  - (e) Referencias.
- (b) La tarea debe ser realizada en `jupyter` notebook (con `Python2` o `Python3`).
- (c) Se evaluará la correcta utilización de librerías `NumPy` y `SciPy`, así como la correcta implementación de algoritmos vectorizados cuando se indique.
- (d) El archivo de entrega debe denominarse `Tarea-1-rol.tar.gz`, y debe contener un directorio con todos los archivos necesarios para ejecutar el notebook, junto con un archivo `README` indicando explícitamente la versión utilizada de `Python`.
- (e) El descuento por día de atraso será de 30 puntos, con un máximo de 1 día de atraso. No se recibirán entregas después de este día.
- (f) El trabajo es personal, no se permite compartir código, aunque sí se sugiere discutir aspectos generales con sus compañeros. En caso de sospecha de copia, se citará a los involucrados a defender sus conocimientos de manera presencial, evaluación cuya nota reemplazará la obtenida en el laboratorio.
- (g) El no seguir estas instrucciones, implica descuentos en su nota obtenida.
- (h) **Fecha de entrega:** Viernes 19 de Agosto por moodle.