



Eliminación Gaussiana, matrices tridiagonales y difusión

Introducción

El objetivo de este trabajo es la implementación y experimentación con algoritmos de eliminación gaussiana y el estudio en particular del caso de matrices tridiagonales y su aplicación en el modelado de un problema de difusión.

A continuación describimos ciertos conceptos que se van a utilizar para realizar las consignas.

Solución de sistemas de ecuaciones

Dado un conjunto de n ecuaciones con n incógnitas podemos escribir la forma matricial del sistema de la siguiente forma:

$$Ax = b$$

donde $A \in \mathbb{R}^{n \times n}$ es una matriz cuadrada, $x \in \mathbb{R}^{n \times 1}$ un vector columna con las incógnitas y $b \in \mathbb{R}^{n \times 1}$ un vector columna con los términos independientes.

El algoritmo de eliminación gaussiana [1] permite resolver el sistema de ecuaciones en los casos que exista una única solución. También permite identificar si la solución no puede ser encontrada ya sea porque el sistema no tiene soluciones o tiene infinitas.

Para comenzar es útil armar una matriz aumentada $A' \in \mathbb{R}^{n \times n+1}$ en la cual se concatena el vector columna b a la derecha de A . Luego, el algoritmo se puede dividir en dos etapas. En una primera, se realiza la eliminación hacia adelante. Se busca obtener una matriz escalonada a partir de la matriz A' . Para esto se realizan las operaciones necesarias desde la primera fila hasta la última para generar valores nulos en la parte inferior izquierda de la matriz. Las operaciones pueden ser: Multiplicar una fila por un escalar no nulo,

Intercambiar de posición dos filas, Sumar a una fila un múltiplo de otra. Se llama pivot al elemento de la matriz que se toma para realizar las operaciones y generar valores nulo debajo de el. En la segunda parte, con la matriz escalonada, se puede despejar una de las incógnitas, y luego sustituir hacia atrás e ir despejando progresivamente las otras incógnitas. La cantidad aproximada de operaciones aritméticas es de orden $O(n^3)$.

Consideramos una versión del algoritmo sin pivoteo, es decir sin permutaciones de filas. Existen casos donde el pivot por defecto vale cero y el algoritmo no termina y no entrega una solución cuando en realidad si existe solución. Si incluimos pivoteo parcial, podemos realizar permutación de filas para poder tomar un pivot distinto de cero avanzar con la primera etapa. En este caso se suele aprovechar que si el valor que se toma como pivot es el más grande dentro de las opciones, se terminan realizando operaciones numéricas con menor error. También existe una versión de pivoteo total, donde realizan permutaciones de columnas para encontrar el mejor pivot.

Sistema tridiagonal

Un sistema de ecuaciones tridiagonal tiene la pinta:

$$a_i x_{i-1} + b_i x_i + c_i x_{i+1} = d_i,$$

$$\begin{bmatrix} b_1 & c_1 & & & 0 \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & \ddots & \\ & & \ddots & \ddots & c_{n-1} \\ 0 & & & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_n \end{bmatrix}. \quad (1)$$

El sistema está definido por a , b , c y d 4 vectores de largo n . Los casos a_1 y c_n se toman como cero. Al verlo de forma matricial se puede apreciar que la mayoría de los elementos de la matriz tienen un valor nulo, y puede considerarse como un caso particular de una matriz rala. A partir de esta observación podemos pensar que si uno aplica el algoritmo de eliminación gaussiana de forma naive va a realizar muchas operaciones redundantes sobre valores nulos.

Si se realiza una versión cuidadosa del algoritmo de eliminación gaussiana utilizando solo los vectores que definen el problema, se puede lograr un algoritmo con menor uso de memoria y con una cantidad de operaciones aritméticas de orden $O(n)$.

La idea es similar a las dos etapas del algoritmo de eliminación gaussiana. Una etapa de eliminación hacia adelante donde se redefinen los vectores b , c y d y luego una sustitución hacia atrás. En la práctica muchas veces ocurre que problemas del tipo $Ax = d$ se realizan de forma sucesiva para distintos casos de d manteniendo A . Si la matriz es tridiagonal, es

ventajoso partir el procesos en 2 etapas, realizar la redefinición de las diagonales b y c una sólo vez, y luego para un dado término independiente d realizar las operaciones que faltan y la substitución hacia atrás.

A continuación un caso de aplicación de una matriz tridiagonal basada en un operador diferencial.

Laplaciano Discreto

El operador de Laplace o Laplaciano (∇^2) es un operador diferencial dado por la divergencia del gradiente de una función escalar. En el caso unidimensional es equivalente a la derivada segunda ($\frac{d^2}{dx^2}$). La versión discreta del operador es ampliamente utilizada procesamiento de imágenes y grafos. Para el caso unidimensional la operación discreta de $\frac{d^2}{dx^2}u = d$ está dada por:

$$u_{i-1} - 2u_i + u_{i+1} = d_i \quad (2)$$

con $u_{-1} = 0$, $u_n = 0$. Notar como la derivada de un valor constante es 0, los coeficientes $(1, -2, 1)$ suman 0. En forma matricial obtenemos:

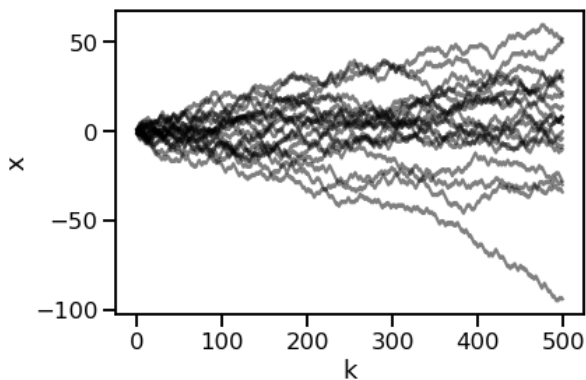
$$\begin{bmatrix} -2 & 1 & & & 0 \\ 1 & -2 & 1 & & \\ & 1 & -2 & \ddots & \\ & & \ddots & \ddots & 1 \\ 0 & & & 1 & -2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_n \end{bmatrix}.$$

Se puede utilizar el algoritmo de eliminación gaussiana para sistemas tridiagonales para encontrar el vector u para cada vector d . Cuando $n \rightarrow \infty$ la solución del sistema de ecuaciones es equivalente a encontrar una función u cuya segunda derivada es la función d (para funciones continuas).

Difusión

La difusión es un proceso estocástico donde una entidad se difunde típicamente desde un lugar de mayor concentración hacia uno de menos. Estos procesos pueden utilizarse para modelar muchos escenarios estocásticos de la naturaleza y se aplican en física, estadística, probabilidad, redes neuronales, finanzas, etc.

Como ejemplo sencillo, en este gráfico se ven varias simulaciones de la evolución temporal de una variable $x^{(k)}$ que sigue la fórmula: $x^{(k)} = x^{(k-1)} + \xi$, donde ξ es una variable aleatoria que toma valores -1 o 1 .



De forma global se puede pensar como que al inicio tenemos una cantidad de partículas en la posición $u^{(0)} = 0$ y durante la evolución estas se difunden de forma aleatoria moviéndose hacia sus regiones aledañas. Una pregunta que se desprende de ver estas trayectorias individuales es ¿cómo será la evolución promedio de una densidad de partículas inicial?

Para resolver esta pregunta, resolveremos la ecuación de difusión de forma discreta utilizando eliminación gaussiana de un sistema tridiagonal [2]. Tendremos un vector inicial $u^{(0)}$ con valores positivos indicando una magnitud que podrá difundirse hacia los costados, reduciendo su valor localmente. Es decir, para cada punto discreto i , u_i se atenuará y se dispersará hacia u_{i-1} y u_{i+1} . Podemos usar el operador laplaciano ya que la suma de sus coeficientes es 0. Consideremos el incremento para el paso k , $u^{(k)} - u^{(k-1)}$ como una fracción (α) del operador laplaciano de u_k :

$$u_i^{(k)} - u_i^{(k-1)} = \alpha(u_{i-1}^{(k)} - 2u_i^{(k)} + u_{i+1}^{(k)}) \quad (3)$$

Reordenando los términos se puede expresar el sistema de forma matricial como $Au^{(k)} = u^{(k-1)}$. Resolviendo el sistema de forma iterativa se puede obtener la evolución del vector u para $k = \{1, 2, 3, \dots, m\}$. El resultado que se obtiene es equivalente al de simular infinitas trayectorias individuales y analizar como se distribuyeron en promedio.

Experimentos relacionados. Máquina de galton: <https://youtu.be/Vo9Esp1yaC8> (<https://youtu.be/Vo9Esp1yaC8>). Difusión de tinturas: <https://youtu.be/eebPrMpKG7c> (<https://youtu.be/eebPrMpKG7c>).

Consignas

1. Eliminación Gaussiana sin pivoteo.
 - a) Implementar una función que acepta la matriz A y el término independiente b y devuelve la respuesta x . Debe devolver una excepción si no puede encontrar la solución.
 - b) Encontrar un ejemplo donde el algoritmo no se puede aplicar pero si hubiese pivoteo si

2. Eliminación Gaussiana con pivoteo.

- a) Implementar EG con pivoteo buscando el pivot parcial por filas con valor más grande
- b) Encontrar un ejemplo donde el algoritmo no se puede aplicar por que el sistema no tiene soluciones o tiene infinitas
- c) Incluir una advertencia en la implementación, cuando haya operaciones que puedan incurrir en error numérico en base a una tolerancia
- d) Encontrar un ejemplo donde el algoritmo entrega un resultado aproximado por error numérico

3. Eliminación Gaussiana para un sistema tridiagonal.

- a) Derivar la formulación de EG para el caso tridiagonal
- b) Implementar una función que toma los vectores a, b, c y d y resuelve el sistema.
- c) Implementar una versión que parte el proceso en dos partes, una precalcula el cómputo sobre b y c y la otra resuelve el sistema para d

4. Tiempos de computo

- a) Calcular los tiempos de cómputo para los casos, eliminación gaussiana con pivoteo, tridiagonal para distintos tamaños de matrices.
- b) Comparar los tiempos para tridiagonal con tridiagonal con precómputo, para distinta cantidad de repeticiones.

5. Derivada segunda

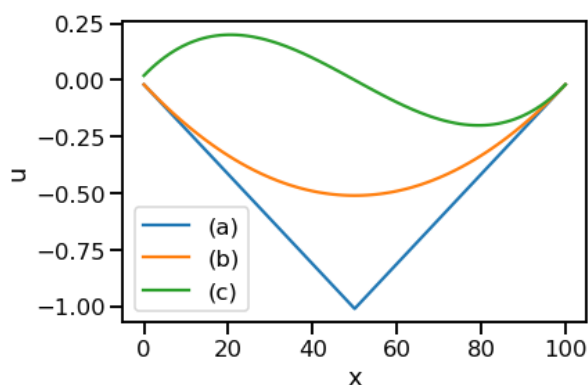
Encontrar u para el problema $\frac{d^2}{dx^2}u = d$ utilizando la matriz tridiagonal del operador laplaciano, para los siguientes d :

a) $d_i = \begin{cases} 0 \\ 4/n \end{cases} \quad i = \lfloor n/2 \rfloor + 1$

b) $d_i = 4/n^2$

c) $d_i = (-1 + 2i/(n - 1))12/n^2$

Usar $n = 101$. Graficar la función u para los tres items y obtener la siguiente figura:



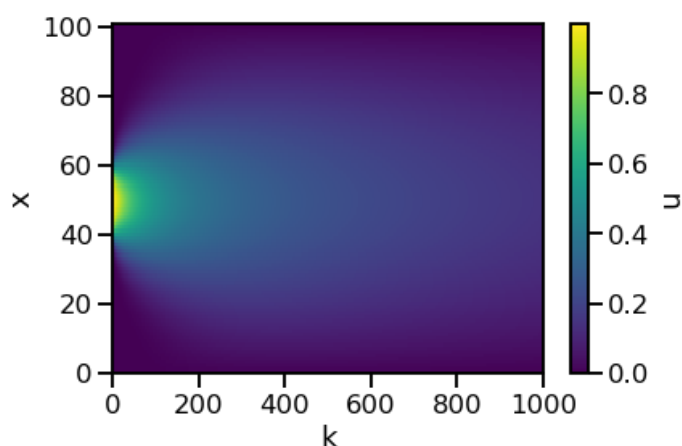
6. Simular difusión

a) Expresar la ecuación (3) como un sistema tridiagonal $Au^{(k)} = u^{(k-1)}$

b) Resolver el sistema de forma iterativa para la condición inicial:

$$u_i^{(0)} = \begin{cases} 0 & \text{if } i \leq 0 \text{ or } i \geq 101 \\ 1 & \text{if } \lfloor n/2 \rfloor - r < i < \lfloor n/2 \rfloor + r \end{cases}$$

Usar $n = 101$, $r = 10$, $m = 1000$ iteraciones. Graficar la evolución de u y generar la siguiente figura.



IMPORTANTE

Las consignas están pensadas para resolverse utilizando python. Para las consignas de implementación de algoritmos se pide que las operaciones de matrices se realicen con python puro, es decir no utilizar numpy ya que se quiere evaluar como se planteó el algoritmo. Solamente se puede utilizar numpy para crear vectores o arreglos. Para el análisis, gráficos y otras cosas se puede usar numpy y otras herramientas.

Forma y Fecha de entrega

1. Se debe entregar un informe hecho en Latex. Este debe incluir una introducción al problema, la solución a las consignas mostrando pseudocódigos con una explicación de su funcionamiento. Figuras o tablas para visualizar resultados junto con sus interpretaciones. Las figuras y tablas tienen que estar referenciadas en el texto y tener una descripción.

Una sección de conclusiones con un resumen y de lo mostrado y los principales resultados.

2. Se debe entregar los códigos que generan los resultados, figuras, etc. Si se requieren procedimientos especiales para ejecutar el código deben ser descriptos.
3. La entrega es el jueves 20 de Abril 23:59 h. Para la entrega se facilitará un formulario online donde deberán adjuntar el informe y los archivos.
4. Recuperatorio: Domingo 21 de Mayo hasta las 23.59 hs, enviando el trabajo corregido.

Referencias

[1] Higham, N. J. (2011). Gaussian elimination. Wiley Interdisciplinary Reviews: Computational Statistics, 3(3), 230-238... [pdf](https://wires.onlinelibrary.wiley.com/doi/pdfdirect/10.1002/wics.164?casa_token=cHtuxpW0tGYAAAAA:st9iQ-vtw2-t8h5e_44oPPukDY1juQMaRR0il5LWcpRkSSFCCMkHg-xJCYg8AVI4J0bGp3hao0TffDy)
(https://wires.onlinelibrary.wiley.com/doi/pdfdirect/10.1002/wics.164?casa_token=cHtuxpW0tGYAAAAA:st9iQ-vtw2-t8h5e_44oPPukDY1juQMaRR0il5LWcpRkSSFCCMkHg-xJCYg8AVI4J0bGp3hao0TffDy).

[2] Langtangen, H. P. (2013). Finite difference methods for diffusion processes. University of Oslo. [pdf](http://ndl.ethernet.edu.et/bitstream/123456789/32700/1/2.pdf.pdf) (<http://ndl.ethernet.edu.et/bitstream/123456789/32700/1/2.pdf.pdf>).