REVISTA DE INVESTIGACIÓN EN MODELOS MATEMATICOS APLICADOS A LA GESTION Y LA ECONOMIA - AÑO 8 VOLUMEN II (2021-II)

CMA IADCOM - UBA CENTRO DE INVESTIGACIÓN EN METODOS CUANTITATIVOS APLICADOS A LA ECONOMÍA Y LA GESTIÓN

Universidad de Buenos Aires Facultad de Ciencias Económicas

http://www.economicas.uba.ar/institutos_y_centros/revista-modelos-matematicos/

UNA ESTRATEGIA DE INNOVACIÓN EN ÁLGEBRA. VECTORES, MATRICES, DETERMINANTES Y SISTEMAS DE ECUACIONES LINEALES CON PYTHON

Fraquelli, Alicia; Gache, Andrea y Gogni, Valeria

Universidad de Buenos Aires. Facultad de Ciencias Económicas. Ciudad Autónoma de Buenos Aires, República Argentina

aliciafraquelli@gmail.com andreagache@gmail.com valeria.gogni@gmail.com

Resumen

Recibido: 18-10-2022

Aceptado: 16-12-2022

Palabras clave

Vectores. Matrices. Determinantes. Sistemas de ecuaciones lineales. Autovalores y Autovectores. Phyton.

El desarrollo de la tecnología sigue un ritmo acelerado y su inclusión en las prácticas educativas no solo es beneficioso sino necesario. Estas tecnologías no solo favorecen la representación matemática múltiple, sino también recursos extraordinarios en la interacción estudiante-conocimiento, que permite un involucramiento activo del sujeto en su aprendizaje.

Por este motivo, con el afán de contribuir a la inserción de la tecnología en la práctica educativa, presentamos la incorporación del lenguaje Python en los contenidos de la materia Álgebra correspondiente al Primer Tramo de las carreras de la Facultad de Ciencias Económicas de la Universidad de Buenos Aires.

El objetivo de esta presentación es compartir el trabajo inicial de rediseño del material de cátedra promoviendo el uso del lenguaje de programación Python.

Una de las implicaciones más importantes del uso de los lenguajes de programación en la enseñanza del álgebra es poder proveer experiencias de aprendizaje que ayuden a desarrollar pensamientos concretos para ayudar en los procesos de abstracción necesarios para el aprendizaje de conceptos matemáticos avanzados.

Las autoras creemos que la integración de los contenidos teóricos y prácticos con Python potencia el aprendizaje, posibilita trabajar con objetos de dimensión mayor a tres evitando cálculos tediosos, permitiendo enfatizar sobre el análisis de los resultados, acercándolos a la realidad de su campo profesional futuro aumentando la motivación de los alumnos.

ISSN (En línea) 2362 3225

AN INNOVATION STRATEGY IN ALGEBRA. VECTORS, MATRICES, DETERMINANTS AND SYSTEMS OF LINEAR EQUATIONS WITH PYTHON

Abstract

The development of technology follows an accelerated pace and its inclusion in educational practices is not only beneficial but necessary. These technologies not only favor multiple mathematical representation, but also extraordinary resources in student-knowledge interaction, which allows an active involvement of the subject in their learning.

For this reason, with the aim of contributing to the insertion of technology in educational practice, we present the incorporation of the Python language in the contents of the subject Algebra corresponding to the First Tranche of the careers of the Faculty of Economic Sciences of the University of Buenos Aires.

The objective of this presentation is to share the initial work of redesigning the teaching material promoting the use of the Python programming language.

One of the most important implications of the use of programming languages in the teaching of algebra is to be able to provide learning experiences that help develop concrete thoughts to help in the abstraction processes necessary for the learning of advanced mathematical concepts.

The authors, we believe that the integration of theoretical and practical content with Python enhances learning, makes it possible to work with objects larger than three avoiding tedious calculations allowing to emphasize on the analysis of the results bringing them closer to the reality of their future professional field increasing the motivation of the students.

Keywords

Vectors. Matrices. Determinants. Systems of linear equations. Eigenvalues and eigenvectors. Python

ISSN (En línea) 2362 3225

INTRODUCCIÓN

El Álgebra lineal es una rama de la matemática que es utilizada en el estudio de una gran variedad de temas vinculados con las Ciencias Económicas, como ser las finanzas, investigación operativa, entre otras.

Es sabido que, por ejemplo, el cálculo matricial y/o la resolución de sistemas de ecuaciones lineales es sin duda tedioso, principalmente cuando el tamaño de las matrices y/o el número de incógnitas aumenta. En esta publicación se pretende mostrar la incorporación de Python para la realización de estas operaciones. De este modo el lector/alumno tendrá la oportunidad de realizar los cálculos que involucren vectores, matrices y resolución de sistemas de ecuaciones lineales cuando lo requiera y como aplicación a lo largo del desarrollo de los contenidos de la materia.

Python, como lenguaje de programación que es, es un entorno adecuado para programar diversos algoritmos, sin embargo, cabe mencionar que no se quiere dedicar la atención al aprendizaje de la programación sino al uso de algunas de las herramientas que Python tiene incorporadas para resolver el cálculo vectorial, matricial y los sistemas de ecuaciones.

No obstante, como docentes sabemos que es importante resaltar la importancia de saber realizar tales operaciones sin la necesidad de una computadora.

1. DESCRIPCIÓN DE LA PROPUESTA

Acordamos con Galindo Alba, (2017) que, en los últimos años, los alumnos que llegan a las aulas universitarias han nacido en un mundo con internet y para ellos es impensable un mundo sin tecnología, están acostumbrados a obtener información con facilidad, tienen capacidad de procesamiento paralelo, son altamente multimediales y aprenden de manera distinta. En este contexto, consideramos que los docentes debemos proponer mejoras en nuestras prácticas, que permitan relacionar nuestros conocimientos con esta nueva generación.

En la actualidad en la que el manejo de software y herramientas informáticas especializadas representa un pilar fundamental en la enseñanza universitaria de grado, posgrado e investigación científica, hemos considerado atinado la incorporación del uso de lenguaje Python para apoyar el desarrollo práctico de los contenidos de la asignatura.

Se plantea como objetivo su uso como una herramienta para el cálculo numérico, no solo en el cursado de la materia Algebra del Primer Tramo de las carreras de la Facultad de Ciencias Económicas (UBA) sino para todo alumno que lo requiera en las materias del Área Matemática que se dictan en la institución en Segundo Tramo, Ciclo Profesional y en el Posgrado.

Esto es especialmente interesante debido a la presencia de contenidos en el programa que se prestan claramente al diseño de algoritmos y a una experimentación mayor que la que permite la enseñanza tradicional.

1.1 Principales beneficios del lenguaje Phyton

Entre los principales beneficios es que es Open Source. El código abierto en el que se basa Python permite utilizar un modelo basado en la comunidad, pudiéndose ejecutar tanto en entornos Windows como Linux. Además de eso, puede ser portado a otras plataformas.

Python se utiliza para la computación científica, tanto a nivel académico como en el mercado de
Python se utiliza para la computación científica, tanto a nivel académico como en el mercado de
trabajo, posee una gran cantidad de bibliotecas de análisis, incluyendo paquetes para el análisis de datos, estadística, visualización y aprendizaje automático

1.2 Google Colab y Librerías de Python para álgebra lineal

Para facilitar la escritura del código en Python se trabajó en el entorno de Google. Basado en Jupyter, es un proyecto de código abierto que permite la ejecución de código Python dentro de un navegador. No requiere configuración, el ambiente está listo para ser usado. Python y muchas librerías están instaladas por defecto. Es intuitivo, por lo que no se necesita conocimiento previo para usar la herramienta.

Existen celdas de texto y de código para mantener documentado el programa desarrollado, además de todas las notas que se pueden agregar aparte de los comentarios del código. Google Colaboratory está enlazado a Google drive, por lo que los blocs de notas se guardan por defecto en este servicio en una carpeta que se crea de manera automática. Las principales librerías que Python ofrece para realizar operaciones de Álgebra lineal son:

NumPy es la biblioteca fundamental para la computación científica en Python. Provee arreglos multidimensionales y una variedad de rutinas para realizar operaciones a gran velocidad, acercándose en rendimiento a los lenguajes compilados. En concreto, algunas de sus funcionalidades son: trabajo con matrices, vectores. La documentación oficial puede ser accedida en línea de manera gratuita.

- **numpy.linalg**: Este es un submódulo dentro de NumPy con un gran número de funciones para resolver ecuaciones de Álgebra lineal.
- **scipy.linalg**: Este submódulo del paquete científico Scipy es muy similar al anterior, pero con algunas más funciones y optimaciones.

SymPy es una biblioteca de Python para matemáticas simbólicas. Es fácilmente extensible, está escrito completamente en Python y no requiere bibliotecas externas. Más allá de su uso como herramienta interactiva, SymPy puede integrarse en otras aplicaciones y ampliarse con funciones personalizadas. Incluye funciones que van desde la aritmética simbólica básica hasta el cálculo, álgebra, matemática discreta y física cuántica

2. PROGRAMACIÓN EN PHYTON

A continuación, se muestra utilizando el Google Colaboratory, la programación en lenguaje Phyton de temas básicos del álgebra lineal a fin de complementar el desarrollo que de los mismos se encuentra en el libro Notas de Álgebra Teórico - Prácticos (Fraquelli - Gache, 2019), siendo el mismo parte de la bibliografía de catedra de los cursos de álgebra de nuestra institución

2.1 Vectores

Vectores con Sympy

En SymPy los vectores se pueden definir mediante la clase **Matrix** del módulo **matrices**

Definir los siguientes vectores $\vec{u}$ y $\vec{v}$ en $\mathbb{R}^3$:

$\vec{u} = \begin{pmatrix} 2 \\ 1 \\ -5 \end{pmatrix}$ y $\vec{v} = \begin{pmatrix} 4 \\ -1 \\ 3 \end{pmatrix}$

.

```
[ ] from sympy import * #Importamos la bilioteca SymPy

[ ] u = Matrix([2,1,-5])
    u
```

$$\begin{bmatrix} 2 \\ 1 \\ -5 \end{bmatrix}$$

```
[ ] v = Matrix([4,-1,3])
    v
```

$$\begin{bmatrix} 4 \\ -1 \\ 3 \end{bmatrix}$$

Operaciones entre vectores

```
[ ] u+v #suma de vectores
```

$$\begin{bmatrix} 6 \\ 0 \\ -2 \end{bmatrix}$$

```
[ ] 2*u #producto por un escalar
```

$$\begin{bmatrix} 4 \\ 2 \\ -10 \end{bmatrix}$$

```
[ ] 5*u+2*v
```

$$\begin{bmatrix} 18 \\ 3 \\ -19 \end{bmatrix}$$

Norma de un vector

Calcular la norma del vector $\vec{v}$.

Como

$$\vec{v} = \begin{pmatrix} 4 \\ -1 \\ 3 \end{pmatrix}$$

entonces $\|\vec{v}\| = \sqrt{4^2 + (-1)^2 + 3^2} = \sqrt{26}$

```
[ ] v.norm() #Norma del vector v
```

$\sqrt{26}$

Producto escalar entre dos vectores

Hallar el producto escalar entre los vectores $\vec{u}$ y $\vec{v}$.

Recordemos que $\vec{u}$ y $\vec{v}$:

$$\vec{u} = \begin{pmatrix} 2 \\ 1 \\ -5 \end{pmatrix} \text{ y } \vec{v} = \begin{pmatrix} 4 \\ -1 \\ 3 \end{pmatrix}$$

El productor escalar resulta:

$$\vec{u}.\vec{v} = 2.4 + 1.(-1) + (-5).3 = -8$$

Veamos el código....

```
[ ] u.dot(v)
```

-8

```
[ ] v.cross(u)   #Producto vectorial entre v y u
```

$$\begin{bmatrix} 2 \\ 26 \\ 6 \end{bmatrix}$$

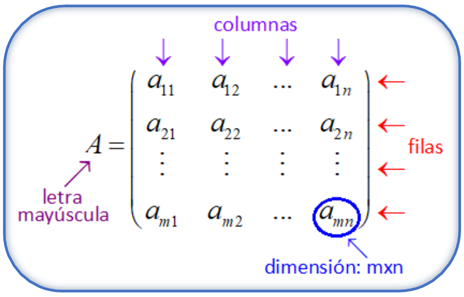
2.2 Matrices

El conjunto de todas las matrices de m filas y n columnas definidas sobre un cuerpo se anota con K^{mxn}

Notación rectangular:

$A = (a_{ij}) \in K^{mxn}$ es la matriz de m filas y n columnas definidas sobre el cuerpo K ($\mathbb{Q}$: racionales, $\mathbb{R}$: reales o $\mathbb{C}$: complejos)

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \in K^{m\,x\,n}$$



Cada elemento posee dos subíndices: el primero "i" hace referencia a la fila que ocupa el mismo en la matriz y el segundo "j" a la columna.

Para su creación, selección de elementos, inserción o eliminación de filas o columnas se utiliza la librería SymPy

Crear las siguientes matrices:

$$A = \begin{pmatrix} 5 & 6 & 2 \\ 4 & 7 & 9 \\ 0 & 3 & 12 \end{pmatrix} \quad B = \begin{pmatrix} 14 & -2 & 12 \\ 4 & 4 & 5 \\ 5 & 5 & 1 \end{pmatrix}$$

```
[ ] A=Matrix([[5, 6,2], [4, 7,9], [0, 3,12]])
    A
```

$$\begin{bmatrix} 5 & 6 & 2 \\ 4 & 7 & 9 \\ 0 & 3 & 12 \end{bmatrix}$$

```
[ ] A.shape  #Permite obtener la dimensión de una matriz
```

(3, 3)

```
[ ] B=Matrix([[14, -2,12], [4, 4,5], [5, 5,1]])
    B
```

$$\begin{bmatrix} 14 & -2 & 12 \\ 4 & 4 & 5 \\ 5 & 5 & 1 \end{bmatrix}$$

```
[ ] b=B[1,2] #Permite seleccionar un elemento de la matriz B
    b
```

5

```
[ ] #Para seleccionar una fila o columna de una matriz utilizamos:
    B.row(0)  #Selecciona la fila 1 de la matriz B
```

$$\begin{bmatrix} 14 & -2 & 12 \end{bmatrix}$$

```
[ ] B.col(-1) #Selecciona la última columna de la matriz B
```

$$\begin{bmatrix} 12 \\ 5 \\ 1 \end{bmatrix}$$

Eliminación de una fila o columna

```
[ ] B=Matrix([[14, -2,12], [4, 4,5], [5, 5,1]])
    # Para eliminar filas o columnas utilizamos row_del o col_del
    B.col_del(0)
    B
```

$$\begin{bmatrix} -2 & 12 \\ 4 & 5 \\ 5 & 1 \end{bmatrix}$$

```
[ ] B=Matrix([[14, -2,12], [4, 4,5], [5, 5,1]])
    B.row_del(2) #Elimina la última fila de la matriz
    B
```

$$\begin{bmatrix} 14 & -2 & 12 \\ 4 & 4 & 5 \end{bmatrix}$$

Inserción de filas o columnas

```
[ ] #  Para insertar filas o columnas utilizamos: row_insert or col_insert
    B=Matrix([[14, -2,12], [4, 4,5], [5, 5,1]])
    B
```

$$\begin{bmatrix} 14 & -2 & 12 \\ 4 & 4 & 5 \\ 5 & 5 & 1 \end{bmatrix}$$

```
[ ] #Insertamos una fila en la matriz B
    B.row_insert(1, Matrix([[1, 1,1]]))
```

$$\begin{bmatrix} 14 & -2 & 12 \\ 1 & 1 & 1 \\ 4 & 4 & 5 \\ 5 & 5 & 1 \end{bmatrix}$$

```
[ ] #Insertamos una columna
    B=Matrix([[14, -2,12], [4, 4,5], [5, 5,1]])
    B
```

$$\begin{bmatrix} 14 & -2 & 12 \\ 4 & 4 & 5 \\ 5 & 5 & 1 \end{bmatrix}$$

```
[ ] B.col_insert (1,Matrix([4, 2, 3]))
```

$$\begin{bmatrix} 14 & 4 & -2 & 12 \\ 4 & 2 & 4 & 5 \\ 5 & 3 & 5 & 1 \end{bmatrix}$$

Operaciones con matrices

Adición de matrices del mismo orden sobre el mismo cuerpo

Sean $\boldsymbol{A=(a_{ij})\in\Re^{mxn}\wedge B=(b_{ij})\in\Re^{mxn}\ :A+B=C}$ siendo $\boldsymbol{C=(c_{ij})\in\Re^{mxn}\ /\ \forall i,\forall j : a_{ij}+b_{ij}=c_{ij}}$

Multiplicación de una Matriz por un Escalar

$$\text{"."}: \Re x \Re^{mxn} \rightarrow \Re^{mxn} \ / \ \forall\alpha\in\Re \ \wedge \ \forall A = (a_{ij})\in\Re^{mxn} : \alpha.A=B \ / \ \forall i , \forall j : \alpha.\, a_{ij} = b_{ij}$$

Multiplicación de Matrices

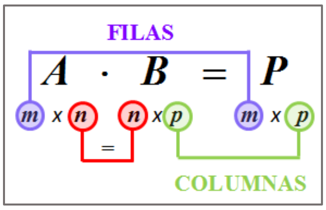
Dadas dos matrices ***A*** y ***B***, su producto es otra matriz ***P*** cuyos elementos se obtienen multiplicando escalarmente los elementos de las filas de ***A*** por los elementos de las columnas de ***B***; los elementos de

P son de la forma:

$$\boxed{p_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}}$$

Para poder realizar la multiplicación, el número de columnas de ***A*** debe coincidir con el número de filas de ***B***.

Si ***A*** tiene orden ***m x n*** y ***B*** orden ***n x p***, la matriz ***P*** será de orden ***m x p***, es decir ***P*** tendrá tantas filas como la matriz ***A*** y tantas columnas como la matriz ***B***.



Operaciones con matrices en SymPy

Dadas las matrices:

$$A = \begin{pmatrix} 2 & 3 \\ -1 & 4 \end{pmatrix}$$

$$B = \begin{pmatrix} 6 & -2 \\ 8 & 4 \end{pmatrix}$$

```
[ ]  A=Matrix([[2, 3], [-1,4]])
     A
```

$$\begin{bmatrix} 2 & 3 \\ -1 & 4 \end{bmatrix}$$

```
[ ]  H=5*A #Multiplicación por un escalar
```

$$\begin{bmatrix} 10 & 15 \\ -5 & 20 \end{bmatrix}$$

```
[ ]  B=Matrix([[6, -2], [8,4]])
     B
```

$$\begin{bmatrix} 6 & -2 \\ 8 & 4 \end{bmatrix}$$

```
[ ]  A+B #Suma de matrices
```

$$\begin{bmatrix} 8 & 1 \\ 7 & 8 \end{bmatrix}$$

```
[ ]  A*B #Producto de matrices
```

$$\begin{bmatrix} 36 & 8 \\ 26 & 18 \end{bmatrix}$$

```
[ ]  C=A**3
     C
```

$$\begin{bmatrix} -16 & 75 \\ -25 & 34 \end{bmatrix}$$

```
[ ]  #Traspuesta de una matriz
     C.T
```

$$\begin{bmatrix} -16 & -25 \\ 75 & 34 \end{bmatrix}$$

Matriz Inversa

Sea $\boldsymbol{A = (a_{ij}) \in \Re^{nxn}}$ es inversible, es no singular, es regular o admite inversa si existe una matriz denominada $\boldsymbol{A^{-1}}$ que multiplicada a izquierda o a derecha por la matriz $\boldsymbol{A}$ da la matriz identidad.

En símbolos: $\boldsymbol{\forall A \in \Re^{nxn}, \; si \; \exists \, A^{-1} \in \Re^{nxn} \Rightarrow A \, . \, A^{-1} = A^{-1} . \, A \; = \; I}$

Importante: A^{-1}*está definida si y solo si* $a.d - c.b \neq 0$

$$\text{Sea } A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathbb{R}^{2x2} \quad \Rightarrow \quad A^{-1} = \frac{1}{|A|} \cdot AdjA^T \quad \Rightarrow \quad \boxed{A^{-1} = \frac{1}{a.d - c.b} \cdot \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}}$$

Hallar la inversa y el determinante de la siguiente matriz: $H = \begin{pmatrix} 4 & -2 \\ 6 & 3 \end{pmatrix}$

```
[ ]  H=Matrix([[4, -2], [6,3]]) #Creamos la matriz H
     H_inv=H**-1
     H_inv
```

$$\begin{bmatrix} \frac{1}{8} & \frac{1}{12} \\ -\frac{1}{4} & \frac{1}{6} \end{bmatrix}$$

```
[ ]  #Verificamos
     H*H_inv
```

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

```
[ ]  H.det() #Calculamos el determinante de la matriz H
```

24

Matrices especiales

- **Matriz Identidad o Unidad**

  Es toda matriz escalar tal que $\boldsymbol{\forall i : \; a_{ii} \; = \; 1}$

  Podemos definir también I es la identidad $\boldsymbol{\Leftrightarrow \forall i, \forall j : \begin{cases} a_{ij} = 1 & si \quad i = j \\ a_{ij} = 0 & si \quad i \neq j \end{cases}}$

$$\boldsymbol{I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}$$

eye(): Permite crea la matriz identidad

```
eye(3) #Creamos la matriz identidad de dimensión 3
```

Matriz Nula

$\boldsymbol{N}$ es la matriz nula $\Leftrightarrow \forall \boldsymbol{i}\ , \forall \boldsymbol{j}\ :\ \boldsymbol{n_{ij}} = \mathbf{0}$

$$\boldsymbol{N} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \in \mathfrak{R}^{3x3} \qquad \boldsymbol{N} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} \in \mathfrak{R}^{3x1} \qquad \boldsymbol{N} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \in \mathfrak{R}^{2x3}$$

zeros(n, m): Permite crear matrices nulas de dimensión nxm

```
zeros(2,3)
```

ones(n, m): Permite crear matrices donde todos sus elementos son 1 de dimensión nxm

Matriz Diagonal

$\boldsymbol{A} = (\boldsymbol{a_{ij}}) \in \mathfrak{R}^{nxn}$ es diagonal $\Leftrightarrow \forall \boldsymbol{i}\ , \forall \boldsymbol{j}\ : \boldsymbol{i} \neq \boldsymbol{j} \Rightarrow \boldsymbol{a_{ij}} = \mathbf{0}$

$$\boldsymbol{A} = \begin{pmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{3} \end{pmatrix} \qquad \boldsymbol{B} = \begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}$$

diag(): Permite crear matrices diagonales

```
diag(4,2,1) #Creamos una matriz diagonal
```

```
eye(3) #Creamos la matriz identidad de dimensión 3
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```
zeros(2,3)
```

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

```
ones(2,4)
```

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

```
diag(4,2,1) #Creamos una matriz diagonal
```

$$\begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2.3 Sistemas de Ecuaciones Lineales

Un sistema de ecuaciones lineales es un conjunto de m ecuaciones lineales con n incógnitas que se expresa de la siguiente forma:

$$\begin{cases} a_{11}\,x_1 + a_{12}\,x_2 + \ldots + a_{1n}\,x_n = b_1 \\ a_{21}\,x_1 + a_{22}\,x_2 + \ldots + a_{2n}\,x_n = b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \ldots \\ a_{m1}\,x_1 + a_{m2}\,x_2 + \ldots + a_{mn}\,x_n = b_m \end{cases}$$

Los números reales $a_{11}, a_{12}, \ldots, a_{1n}$ en general a_{ij} son los coeficientes de las incógnitas.

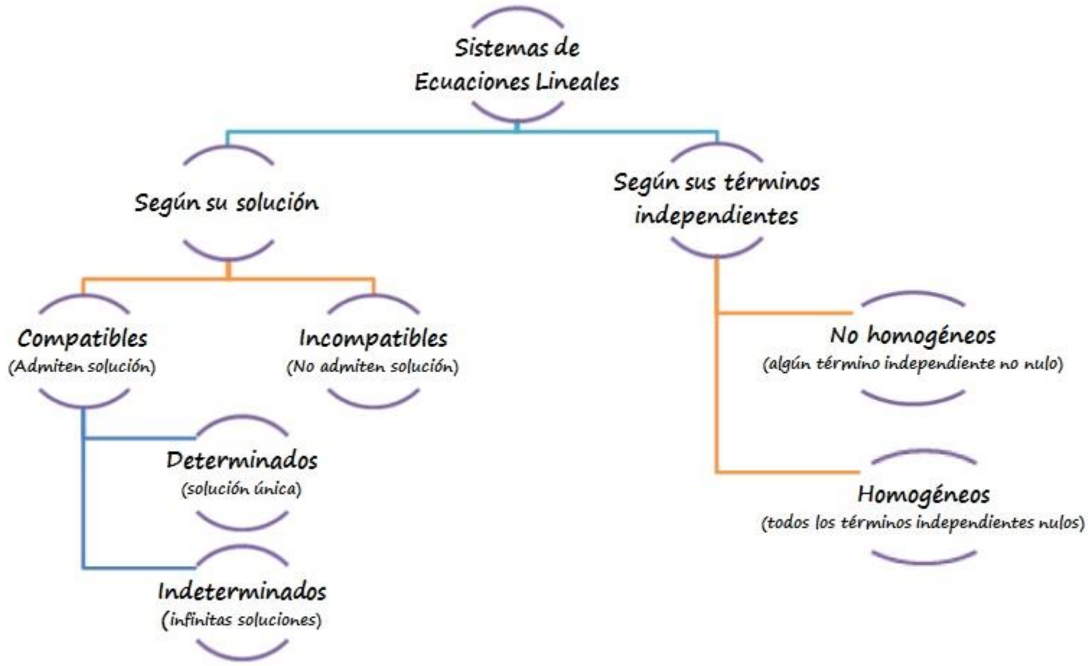
Las incógnitas del sistema están representadas por $x_1, x_2, \ldots, x_j, \ldots, x_n$.

Los números reales $b_1, b_2, \ldots, b_i, \ldots, b_m$ son los términos independientes del sistema.

Solución del sistema: es el conjunto de valores de la n-upla $(x_1; x_2; \ldots; x_n)$ que satisfacen simultáneamente a todas y cada una de las ecuaciones del sistema.

Conjunto solución: es el conjunto formado por todas las soluciones del sistema.

Clasificación de los sistemas de ecuaciones lineales en función de su conjunto solución y sus términos independientes.



A partir de los siguientes ejemplos se muestra el código a utilizar para resolver sistemas de ecuaciones lineales y las diferentes posibilidades de solución.

1er. Caso: Sistema Compatible Determinado (SCD)

Resolver el siguiente sistema de ecuaciones:

$$\begin{cases} 3x + 2y = 5 \\ x - 3y = -2 \end{cases}$$

```
[2]  x=Symbol('x') #Definimos las variables x e y
     y=Symbol('y')

[3]  solve([3*x+2*y-5,x-3*y+2],[x,y])

     {x: 1, y: 1}
```

2do. Caso: Sistema Compatible Indeterminado (SCI)

Resolver el siguiente sistema de ecuaciones:

$$\begin{cases} 3x - 7y = 1 \\ -6x + 14y = -2 \end{cases}$$

```
[12] x=Symbol('x') #Definimos las variables x e y
     y=Symbol('y')
```

```
[19] solve([3*x-7*y-1,-6*x+14*y+2],[x,y])

     {x: 7*y/3 + 1/3}
```

3er. Caso: Sistema Incompatible (SI)

Resolver el siguiente sistema de ecuaciones:

$$\begin{cases} 4x + 3y = 18 \\ 5x - 6y = 3 \end{cases}$$

```
[ ] x=Symbol('x') #Definimos las variables x e y
    y=Symbol('y')
```

```
[ ] solve([3*x+3*y-1,x+y-2],[x,y])

    []
```

3.4 Autovalores y Autovectores

Sea $A \in \mathbb{R}^{nxn}$, $\lambda \in \mathbb{R}$ es autovalor o valor propio o valor característico de la matriz $\boldsymbol{A}$ si y sólo si existe un vector $\vec{v} \in \mathbb{R}^{nx1}$ no nulo tal que:

$$\boldsymbol{A}.\vec{\boldsymbol{v}} = \boldsymbol{\lambda}\vec{\boldsymbol{v}} \quad \boldsymbol{con}\,\vec{\boldsymbol{v}} \neq \vec{\boldsymbol{0}}$$

Operando se obtiene:

$$(\boldsymbol{A} - \boldsymbol{\lambda I})\vec{\boldsymbol{v}} = \vec{0}$$

Las correspondientes soluciones de la ecuación para cada valor de λ se denomina autovector o vector propio o vector característico de la matriz $\boldsymbol{A}$

Cálculo de valores y vectores propios

La búsqueda y cálculo de valores propios para una matriz $\boldsymbol{A}$ se apoya en las equivalencias siguientes:

$$\lambda \text{ es un valor propio de } A$$

$$\Leftrightarrow A.\vec{v} = \lambda\vec{v} \quad \text{tiene soluciones no nulas}$$

$$\Leftrightarrow (A - \lambda\, I).\vec{v} = \vec{0} \text{ tiene infinitas soluciones}$$

$$\Leftrightarrow det\ (A - \lambda\, I) = \ 0$$

Para hallar autovalores y autovectores de una matriz utilizamos:

Es posible obtener un *dictionary* de la forma **eigenvalue: algebraic_multiplicity**

eigenvals (): Autovalores

Obtener los autovalores de la matriz: $M = \begin{pmatrix} 6 & 1 \\ 4 & 3 \end{pmatrix}$

```
[ ] M=Matrix([[6, 1], [4,3]]) #Creamos la matriz M
    M
```

$\begin{bmatrix} 6 & 1 \\ 4 & 3 \end{bmatrix}$

```
[ ] M.eigenvals() #Obtenemos los autovalores y su multiplicidad
```

```
{7: 1, 2: 1}
```

```
[ ] M.eigenvects() #Obtenemos los autovectores
```

```
[(2, 1, [Matrix([
   [-1/4],
   [   1]])]), (7, 1, [Matrix([
   [1],
   [1]])])]
```

2.5 Diagonalización de matrices

Una matriz A es diagonalizable si existe una matriz inversible P y una matriz diagonal D tal que $A\ =\ PDP^{-1}$

Donde A es la matriz a diagonalizar, P es la matriz cuyas columnas son los vectores propios (o autovectores) de A, P^{-1} su matriz inversa y D es la matriz diagonal formada por los valores propios (o autovalores) de A.

NOTA: Son expresiones equivalentes $\boldsymbol{A\ =\ PDP^{-1}}$ y $\boldsymbol{D\ = P^{-1}AP}$

Diagonalizar la siguiente matriz: $A = \begin{pmatrix} 3 & 1 \\ 2 & 2 \end{pmatrix}$

```
[ ] A=Matrix([[3, 1], [2,2]]) #Creamos la matriz A
    A
```

$\begin{bmatrix} 3 & 1 \\ 2 & 2 \end{bmatrix}$

```
[ ] P, D = A.diagonalize()  #Obtenemos la matriz P
    P
```

$\begin{bmatrix} -1 & 1 \\ 2 & 1 \end{bmatrix}$

```
[ ] D
```

$\begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$

```
[ ] P*D*P**-1
```

$\begin{bmatrix} 3 & 1 \\ 2 & 2 \end{bmatrix}$

Polinomio característico

Es posible obtener el polinomio característico de una matriz utilizando**: charpoly**

```
[ ]  lamda = symbols('lamda')
     p = A.charpoly(lamda)
     factor(p.as_expr())
```

$(\lambda - 4)(\lambda - 1)$

```
[ ]  A.eigenvals() #Verificamos
```

```
{4: 1, 1: 1}
```

CONCLUSIONES

Se necesita una intencionalidad educativa en el uso de Python, más allá de la simple utilización encaminada a mejorar el proceso de enseñanza - aprendizaje haciéndolo más interactivo, más innovador, sobre todo más motivador para el alumno, adaptándolo a la nueva realidad social, cultural y tecnológica que vivimos.

A pesar de las ventajas Python en la enseñanza de los temas de álgebra, es necesario para utilizar el Google Colaboraty y las librerías contar con sesiones de práctica para familiarizarse con el entorno.

Las TIC han abierto muchas alternativas, tanto en la enseñanza como en el aprendizaje de la nueva generación "digitalmente" culturizada, las que, integradas en un entorno o ambiente de aprendizaje con diferente grado de virtualización, ponen a disposición de los docentes canales de información y comunicación para promover formas distintas de enseñanza y es nuestro objetivo aprovecharlas para fortalecer el proceso y lograr alumnos con capacidad de alcanzar aprendizaje crítico y significativo.

Posibilitan la profundización de conocimientos en el quehacer educativo; constituyen un medio excelente para cuestionar ciertas prácticas pedagógicas que se realizan en el aula; incrementan notablemente la participación y la interacción de los alumnos, logrando su integración en situaciones de aprendizaje

Se propicia entonces un uso de la tecnología y las diferentes herramientas digitales enfocadas hacia la participación del alumno, para que éste sea capaz de crear y construir su conocimiento, evaluar su aprendizaje y elaborar sus propias herramientas de análisis de la realidad.

Sin embargo, se debe tener cuidado en la utilización de estos recursos, para que éste sea pedagógico, y no sea una necesidad exclusivamente para resolver una operación matemática.

Se acuerda con Coll (2011) que las instituciones de educación superior vienen experimentando cambios importantes con el objetivo de promover experiencias innovadoras en los procesos de enseñanza aprendizaje apoyándose en las TIC y nuestra Institución no escapa a ese propósito.

REFERENCIAS

ASCHER, D, DUBOIS, P, HINSEN, K, HUGUNIN, J & OLIPHANT, T (2001). Numerical python.

COLL, C. (2011). *Aprender y enseñar con las TIC: expectativas, realidad y potencialidades*

FRAQUELLI, ALICIA D.; GACHE, ANDREA. (2019). *Notas de álgebra teórico-prácticas: Cátedra de álgebra*. Recuperado de http://bibliotecadigital.econ.uba.ar/download/libros/Fraquelli-Gache_Notas-Algebra-Teorico-Practicas.pdf

GALINDO ALBA, A. (2017). *Didáctica con R. Menos cuentas y más pensamiento crítico*. Pensamiento Matemático, Vol. VII (1), 53-74.

KURGALIN, S & BORZUNOV, S (2021). *Algebra and geometry with Python* (pp. 1-425). Springer.