TIPO: CURSO DE POSGRADO.

NOMBRE: Modelado y Simulación: una introducción a la computación científica usando MATLAB.

VIGENCIA: Segundo Cuatrimestre, 2024.

DOCENTES RESPONSABLES: Dr. Ing. Sergio Preidikman y Dr. Ing. Emmanuel Beltramo
Oficina 109; Departamento de Estructuras; Facultad de Ciencias Exactas, Físicas y Naturales; Universidad Nacional de Córdoba; TE: (0351) 433-4145 Interno 35; e-mails: spreidikman@unc.edu.ar

PROFESIONALES A LOS QUE ESTÁ ORIENTADO: Ingenieros Mecánicos, Aeronáuticos, Civiles, Electromecánicos, Electricistas, y Químicos. Licenciados en Matemática, Física y Química. Todo aquel profesional interesado en el Método de los Elementos Finitos y que satisfaga los requisitos descriptos en el ítem "PRE-REQUISITOS".

PRE-REQUISITOS: Tener conocimientos, a nivel de carrera de grado, en: Álgebra Lineal, Cálculo de Varias Variables, Métodos Numéricos, Ecuaciones Diferenciales Ordinarias y Ecuaciones Diferenciales en Derivadas Parciales. Es recomendable tener experiencia previa en programación utilizando MATLAB®. Se entiende que lo que se presenta como "recomendable" no es excluyente. Sin embrago, el postulante que no posea esta experiencia previa deberá realizar un esfuerzo adicional, habida cuenta de la íntima relación con la temática de la asignatura. La aceptación de aquellos postulantes que no satisfagan alguna de las condiciones citadas anteriormente, quedará a criterio de los docentes responsables del dictado de la asignatura, previa evaluación de los antecedentes del postulante.

JUSTIFICACIÓN: Como el título lo sugiere, este curso es una introducción al campo de modelado y simulación en las ciencias y en la ingeniería utilizando MATLAB, como entorno de desarrollo integrado y lenguaje de programación propio, y a los métodos numéricos como las herramientas para hacer computación científica. Este no es un curso de programación en MATLAB.

El enfoque es simple. Cada clase comienza con el planteo de un problema que apunta a una historia computacional más grande. La solución de dicho problema es cuidadosamente derivada y a lo largo del proceso se va introduciendo lo que sea requerido de MATLAB. A esto le sigue una breve charla que apunta a enfatizar algunos aspectos del contexto/problema más general. Este patrón entra en resonancia con lo que creo acerca de lo que debe ser un primer curso sobre modelado y simulación, esto es, debería ser enseñado mediante el uso de ejemplos. Cada clase culmina con la producción de un guión de trabajo de MATLAB y generalmente de unas cuantas funciones de MATLAB. Los ejercicios incluyen problemas directos "*m*-problems" que se enfocan en el desarrollo de códigos y todo lo nuevo en MATLAB. Por su parte, los problemas "*p*-problems" son designados para reforzar el mensaje computacional de los temas cubiertos en cada clase.

Se elige el entorno de programación MATLAB porque es amigable para los programadores principiantes y porque permite jugar con ideas computacionales mediante experimentación. Además, de tratarse del lenguaje de computación científica utilizado por prácticamente todas las universidades y centros de investigación y desarrollo del mundo. Esto es central y fundamental para desarrollar la intuición computacional.

Jugar con programas construye la intuición computación

La intuición es un sentido de la orientación no diferente de aquel que le permite a uno encontrar un camino en un barrio de la infancia pasada sin un mapa. Si la intuición es un sentido de la orientación, entonces la intuición computacional es un sentido de la orientación computacional.

OBJETIVOS: Por medio de ejemplos y problemas se intenta lograr lo siguiente:

1. *Desarrollar ojos para lo geométrico*. La habilidad para visualizar es muy importante para el científico computacional. Por supuesto, los gráficos en computadoras juegan un tremendo rol en este aspecto, pero las herramientas de visualización que muchos programas ofrecen no hacen obvia la necesidad de razonar en términos geométricos. *Resulta muy crítico familiarizarse con senos y cosenos, polígonos y poliedros, métrica y proximidad, etc*.

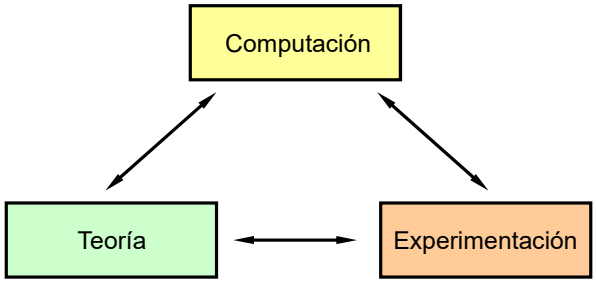
2. *Desarrollar un oído capaz de oír la "explosión combinatoria*." Muchos diseños y problemas de optimización involucran grandes espacios de búsqueda con un número exponencial de posibilidades. *Es importante anticipar esta complejidad y tener los medios para manejarlos con heurísticas inteligentes*.

3. *Desarrollar un gusto por el azar*. La ciencia y la ingeniería están colmadas de procesos que tienen una componente aleatoria. *Tener un sentido de la probabilidad y la capacidad de reunir e interpretar las estadísticas con la computadora es vital*.

4. *Desarrollar un olfato por la dimensión*. La simulación es mucho más intensa computacionalmente en tres dimensiones que en dos — una dura realidad que está mirando a muchos científicos a la cara. Una impresión exacta de cómo las computadoras pueden facilitar la comprensión del mundo físico requiere una apreciación de este punto. Más aun, *ser capaz de pensar a nivel de matriz es esencial para un eficaz y alto rendimiento computacional*.

5. *Desarrollar un toque por lo que es finito, inexacto, y aproximado*. Los errores de redondeo involucran a la aritmética de punto flotante, las pantallas de ordenador son granulares, las derivadas analíticas se aproximan con diferencias finitas divididas, un polinomio se utiliza en lugar de la función seno, y los datos adquiridos en un laboratorio sólo puede ser correctos (con mucha suerte) hasta los tres dígitos significativos. La vida en la ciencia computacional es parecida a esto por lo cual el profesional debe ser lo suficientemente osado como para hacer frente a tales incertidumbres. *Un juego constante de piernas es necesario para mantener el balance en la barra de equilibrio que separa lo continuo de discreto*.

Si bien el desarrollo de estos cinco sentidos es una prioridad explícita, mi ambición primordial es comunicar la importancia de hacer computación al tiempo de saber apreciar sus limitaciones y conocer sus conexiones con otras metodologías. La interacción entre la informática, la teoría y la experimentación es particularmente importante.



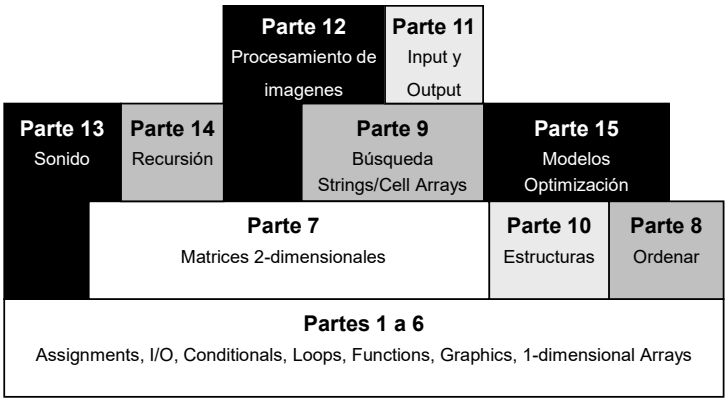
Cada vértice representa un estilo de investigación y proporciona una ventana a través de la cual podemos ver a la ciencia y a la ingeniería como un todo. La vitalidad de lo que vemos en el interior del triángulo depende de las ideas que fluyen alrededor de los bordes. Una buena teoría expresada en lenguaje matemático puede ser materializada en forma de un programa de computación, tal vez sólo para afirmar su exactitud. "Correr" el programa es una simulación que puede sugerir un experimento de física. El experimento puede a su vez revelar un parámetro faltante en el modelo matemático subyacente, y así se inicia el círculo nuevamente.

Hay también dinámicas interesantes en la otra dirección. Un experimento físico puede ser restringido en su alcance por razones de presupuesto o de seguridad, por lo que la escena cambia y pasa a ser una simulación computacional. Lo más probable es que el acto de escribir el programa para realizar la simulación tenga una influencia clarificadora, estimulando alguna nueva actividad matemática. Modelos innovadores son descubiertos, lo que conduce a la modificación de la primera serie de experimentos, y así sucesivamente.

Al pensar en estas interacciones críticas recuerdo al gran científico matemático Richard Hamming, quien declaró en 1960 que "el propósito de la computación es la comprensión, no los números". Estoy evidentemente de acuerdo con este punto de vista. El mensaje que debe quedar de un primer curso de programación debe ser "Comprensión a través de la computación" en lugar de "Salida a través de la informática".

Temas de Programación

Los temas de este curso fácilmente se asocian a los temas de programación de un típico curso de introducción a la programación. La figura de abajo muestra la organización por temas de programación. Cada bloque depende del/de los bloque(s) inmediatamente por debajo. Las partes 1-7 cubren los fundamentos de la programación (bloques blancos), incluyendo el flujo de control, funciones, y matrices de una y dos dimensiones. Las partes 8, 9, y 14 presentan temas/ejemplos algorítmicos reales (bloques de color gris oscuro), tales como clasificación, bisección, y recursividad. Datos sobre estructuras, cadenas y procesamiento de archivos (bloques grises claros) se discuten en las partes 9, 10, y 11. Aplicaciones y ejemplos más avanzados aparecen en las partes 12, 13, y 15 (bloques negros).



DURACIÓN Y ORGANIZACIÓN: 60 horas. Un total de 15 semanas de clases.
El total del tiempo se dedicará a clases teórico-prácticas, las que se desarrollarán en aula. El presente curso no requiere de trabajos de campo, gabinete o laboratorio, visitas o viajes de estudio.

Lugar de Clases: Aula 705, Edificio de Graduados.

Horario: martes de 14:00 a 18:00hs.

Fecha de inicio: 6 de agosto de 2024.

Fecha de finalización: 19 de noviembre de 2024.

METODOLOGÍA DE DICTADO: Se dictará una clase teórica/práctica semanal de 4 (cuatro) horas de duración. La misma se desarrollará en aula, con exposición oral, uso de pizarrón, de proyector multimedia, y de computadoras.

Se fijarán horarios de consulta, fuera del horario de clases, de acuerdo con las posibilidades de los docentes responsables de la asignatura y de los estudiantes inscriptos.

METODOLOGÍA DE EVALUACIÓN Y APROBACIÓN:
La evaluación se realizará mediante: *i*) la presentación de trabajos prácticos y *ii*) la confección de programas de computadora (acompañados de sus respectivos informes) que servirán para implementar los conocimientos adquiridos en el curso.

Todos los trabajos prácticos, proyectos de computadora e informes deberán ser realizados de manera individual por cada estudiante. Los trabajos prácticos y los informes deberán ser entregados en la fecha estipulada por los docentes. No se aceptarán trabajos prácticos e informes tardíos.

La nota final se calculará de la siguiente manera:

- Trabajos prácticos: 50% de la nota final.
- Programas de computadora = 50% de la nota final.

La asistencia a clases no es un requisito para aprobar el curso, aunque la misma es altamente recomendada.

NECESIDADES DE INFRAESTRUCTURA: Disponibilidad de un aula con pizarra blanca, computadoras con MATLAB instalado y un retroproyector.

PROGRAMA ANALÍTICO RESUMIDO:

1. De la fórmula al programa
2. Límites y error
3. Aproximación con fracciones
4. Lo discreto versus lo continuo
5. Abstracción
6. Aleatoriedad
7. La segunda dimensión
8. Reordenamiento
9. Búsqueda
10. Puntos, polígonos y círculos
11. Procesamiento de archivos de texto
12. La matriz: Parte II
13. Procesamiento de archivos acústicos
14. Divide y reina
15. Optimización

PROGRAMA ANALÍTICO DESARROLLADO:

Parte I – DE LA FORMULA AL PROGRAMA

¡Solo conéctalo! (Superficie de una esfera – MATLAB: Expresiones aritméticas, asignación, entrada, salida).
Corrobora y evalúa (Mínimo de una cuadrática en un intervalo – MATLAB: expresiones Booleanas, condicionales).
(MATLAB: The Desktop, Interaction and Script Files, Variables and the Workspace, Scripts and Functions, Working with `m`-Files and the MATLAB).

Parte II – LIMITES Y ERROR

Colocando mosaicos sobre un disco (Sumatoria – MATLAB: "The `for` loop").
Polígonos de adentro y de afuera (Secuencias – MATLAB: "The `while` loop").
(MATLAB: Automatic Storage Allocation, Variable Arguments Lists, Complex Arrays and Arithmetic).

Parte III – APROXIMACION CON FRACCIONES

22/7ths y Contando… (Aproximación a Pi – MATLAB: "loops" anidados, puntos de referencias).
No del todo perfecto (Cocientes de Fibonacci y la Relación de Oro – MATLAB: "More complicated `while`-loops").
(MATLAB: Matrix Generation, Subscripting and the Colon Notation, Matrix and Array Operations, Matrix Manipulation, Empty Matrices, Data Analysis).

Parte IV – LO DISCRETO VERSUS LO CONTINUO

Conecta los puntos (Funciones continúas a trazado – MATLAB: Vectores, gráficos elementales).
Del color turquesa (cyan) al magenta (Computación con colores – MATLAB: `rgb`).
Un tercio más un tercio no es igual a dos tercios… (El mundo del punto flotante – MATLAB: `inf`, `NaN`).
(MATLAB: IEEE Arithmetic, Precedence, Mathematical Functions, Other Data Types, `eps`, `realmin`, `realmax`, `single`, `double`).

Parte V – ABSTRACCION

Dándole nueva forma a los rectángulos (Un cuadrado y una raíz – MATLAB: Funciones simples). El odómetro oval (Perímetro de una elipse – MATLAB: Funciones con parámetros de entradas múltiples).
El problema de Betsy Ross (Parámetros de diseño – MATLAB: Funciones gráficas).
(MATLAB: Function Handles, Anonymous Functions, Local Functions, Variable Numbers of Arguments, Argument Checking and Parsing, Nested Functions, Private Functions, Recursive Functions, Global and Persistent Variables).

Parte VI – ALEATORIEDAD

Seguridad en números (Simulación de Monte Carlo – MATLAB: Más práctica con expresiones Booleanas).
El dado y la brújula (Caminatas aleatorias – MATLAB: Más práctica con "`while` loops").
El orden a partir del caos (Polígono promedio – MATLAB: Más práctica con gráficos y vectores).
(MATLAB: Two-Dimensional Graphics, Basic Plots, Axes and Annotation, Objects and Properties, Multiple Plots in a Figure, Three-Dimensional Graphics, Specialized Graphs for Displaying Data).

Parte VII – LA SEGUNDA DIMENSION

LA SEGUNDA DIMENSION
De aquí hasta allá… (Matrices de transición – MATLAB: Matrices de dos dimensiones).
Contornos y secciones transversales (Visualización de $F(x,y)$ – MATLAB: Trazado de contornos).
¡Deténganse! (Simulación en una red – MATLAB: `A(i,j)` actualizaciones).
(MATLAB: Character Vectors and Arrays, String Arrays, Categorical Arrays, Datetime and Duration Arrays, Tables and Timetables).

Parte VIII – REORDENAMIENTO

Negociar (La combinación perfecta – MATLAB: Mas practica con vectores y subíndices).
Magnitud y ubicación (Clasificación – MATLAB: `sort`).
(MATLAB: Empty Arrays, Exploiting Infinities, Permutations, Rank 1 Matrices, Set Operations).

Parte IX – BUSQUEDA

Patrones de proteínas (Búsqueda lineal – MATLAB: Matrices de caracteres).
Una guía telefónica con números romanos (Búsqueda binaria – MATLAB: Arreglos de celdas).
Cambio de signo (Bisección para encontrar raíces – MATLAB: Funciones como parámetros).
(MATLAB: Multidimensional Arrays, Structures and Cell Arrays).

Parte X – PUNTOS, POLIGONOS Y CIRCULOS

¿Cuán distante? (Métrica de la distancia – MATLAB: Estructuras simples).
¿Cercado en dos ocasiones? (Intersección – MATLAB: Estructuras más complicadas, Funciones Booleanas).
¿Imperfecto? (Cercanía en la forma – MATLAB: Práctica con estructuras).
(MATLAB: Errors and Assertions, Warnings, Debugging, Pitfalls).

Parte XI – LA SEGUNDA DIMENSION

Latitud y luz del día (Adquisición de datos y conversión – MATLAB: Leyendo datos desde un archivo de texto).
Del orden de los millones… (Escritura y representación – MATLAB: Creando archivos `.dat` y archivos `.bin`).
(MATLAB: Objects, Handles, and Properties, Root and Default Properties, Animation, Undirected Graphs, Directed Graphs).

Parte XII – LA MATRIZ: PARTE II

Creando arcoíris privados… (Interpolación lineal y mapeo de color – MATLAB: configuración de matriz fila por fila).
Conocido en la esquina (Interpolación bilineal y sombreado – MATLAB: De $F(x,y)$ a `F(i,j)`).
Siete por Cinco… (Digitalización de imágenes – MATLAB: Arreglos de celdas de matrices).

Imagínate esto… (Trabajando con archivos de datos de imagen – MATLAB: `imread`, `imwrite`, más práctica con matrices).
(MATLAB: Subscripting Matrices as Vectors, Avoiding `if` Statements, Triangular and Symmetric Matrices).

Parte XIII – PROCESAMIENTO DE ARCHIVOS ACUSTICOS

¿Qué hora indican las campanas del reloj? (Adquisición y reproducción – MATLAB: `wavread`, `sound`, `wavwrite`).
Disca N para escuchar un ruido (Frecuencia y muestreo – MATLAB: Más práctica con vectores).
(MATLAB: Timing Code, Vectorization, Accessing Matrices by Column, Preallocating Arrays)

Parte XIV – DIVIDE Y REINA

Patrones dentro de otros patrones (Mosaicos recursivos – MATLAB: Funciones recursivas).
N y la mitad *N* (Árboles binarios – MATLAB: Más práctica usando recursión).
Buscando problemas (Interpolación adaptativa – MATLAB: Más práctica usando recursión).
(MATLAB: Object-Oriented Programming, Max-Plus Algebra Class, Circulant Matrix Class).

Parte XII – OPTIMIZACION

La ruta más corta (Explosión combinatoria – MATLAB: Más práctica con matrices).
La mejor bicicleta (Funciones objetivo y restricciones – MATLAB: "Loops" anidados más complejos).
La órbita más probable (Construcción de modelos – MATLAB: Búsqueda interactiva).
(MATLAB: The Parallel Computing Toolbox, The `parfor` loop, Asynchronous Computing with `parfeval`, Batch Computations, Distributed and Codistributed Arrays, Miscellaneous Optimizations).

BIBLIOGRAFÍA:

1. Van Loan, C. F. and Fan K.-Y. D., *Insight through Computing: A MATLAB Introduction to Computational Science and Engineering*, Society for Industrial and Applied Mathematics, Philadelphia, PA, U.S., ISBN: 978-0-898716-91-7, 2010.
2. Higham, D. J. and Higham, N. J., *MATLAB Guide*, 3rd Edition, SIAM, ISBN: 978-1-6119746-52, 2017.
3. Peter M. Kogge, *The Zen of Exotic Computing*, SIAM, ISBN:978-1-61197-728-8, 2022.
4. Altman, Y., Accelerating MATLAB Performance: 1001 tips to speed up MATLAB programs, CRC Press, ISBN: 978-1-4822-1130-6, 2015.
5. Ionut Danaila; Pascal Joly; Sidi Mahmoud Kaber; Marie Postel, *An Introduction to Scientific Computing: Fifteen Computational Projects Solved with MATLAB*, Springer International Publishing, ISBN: 978-3-031350-32-0, 2023.

REFERENCIAS PARA LECTURA ADICIONAL:

1. Paul A. Gagniuc, *Coding Examples from Simple to Complex: Applications in MATLAB*, Springer, ISBN: 978-3-031538-04-9, 2024.
2. Paul Curzon, Peter W. McOwan, Conjuring with Computation: A Manual of Magic and Computing for Beginners, World Scientific Publishing, ISBN: 978-9-811264-33-7, 2023.
3. Dingyü Xue and Feng Pan, *MATLAB and Simulink in Action: Programming, Scientific Computing and Simulation*, Springer Nature Singapore, ISBN: 978-9-819911-76-9, 2024.
4. Johan Hoffman, *Methods in Computational Science*, SIAM, ISBN:978-1-61197-671-7, 2021.
5. Van Loan, C. F., *Introduction to Scientific Computing: A Matrix-Vector Approach Using MATLAB*, Prentice Hall, NJ, 2000.
6. Gilbert Strang, *Linear Algebra and Learning from Data*, SIAM, ISBN:978-0-692-19638-0, 2019.
7. John Lawrence Nazareth, *Concise Guide to Numerical Algorithmics: The Foundations and Spirit of Scientific Computing*, Springer Briefs in Computer Science, ISBN: 978-3-03121-761-6, 2023.

8. Carl D. Meyer, *Matrix Analysis and Applied Linear Algebra*, Second Edition, SIAM, ISBN:978-1-61197-743-1, 2023.
9. Zhong-Zhi Bai and Jian-Yu Pan, *Matrix Analysis and Computations*, SIAM, ISBN:978-1-61197-662-5, 2021.
10. Moler, C. B., *Numerical Computing with MATLAB*, SIAM, ISBN: 978-0-898716-60-3, 2004.
11. Gander, W., Gander, M. J., and Kwok, F., *Scientific Computing: An Introduction using Maple and MATLAB*, Springer International Publishing Switzerland, ISBN: 978-3-319-04324-1, 2014.
12. Heat, M. T., *Scientific Computing: An Introductory Survey*, Revised Second Edition, SIAM, ISBN:978-1-61197-557-4, 2018.
13. Chapra, S. C., *Applied Numerical Methods with MATLAB*, 4th Edition, McGraw Hill-Education, ISBN: 978-0-07-339796-2, 2018.
14. Tobin A. Driscoll and Richard J. Braun, *Fundamentals of Numerical Computation*, SIAM, ISBN:978-1-61197-507-9, 2017.
15. Lloyd N. Trefethen, Ásgeir Birkisson, and Tobin A. Driscoll, *Exploring ODEs*, SIAM, ISBN:978-1-61197-515-4, 2017.
16. Xue, J., and Chen, Y-Q, *Scientific Computing with MATLAB*, 2nd Edition, CRC Press, ISBN: 978-1-4987-5778-2, 2016.
17. Higham, N. J., *Functions of matrices: theory and computation*, SIAM, ISBN: 978-0-89871-646-7, 2008.
18. Higham, N. J., *Accuracy and Stability of Numerical Algorithms*, 2nd Edition, SIAM, ISBN: 0-89871-521-0, 2002.
19. Chen, K., Giblin, P., and Irving, A., *Mathematical Explorations with MATLAB*, Cambridge University Press, UK, ISBN: 978-0-52-163078-8, 1999.
20. Attaway, S., *MATLAB: A Practical Introduction to Programming and Problem Solving*, 4th Edition, Elsevier Inc., ISBN: 978-0-12-804525-1, 2017.
21. Rouson, D., Xia, J., and Xu, X., *SCIENTIFIC SOFTWARE DESIGN: The Object-Oriented Way*, Cambridge University Press, ISBN: 978-0-521-88813-4, 2011.
22. Acton, F. S., *Real Computing Made Real: Preventing Errors in Scientific and Engineering Calculations*, Princeton University Press, ISBN 0-691-03663-2, 1996.
23. Stephan, E., and Wriggers, P. (Eds.), *Modelling, Simulation and Software Concepts for Scientific-Technological Problems*, Springer-Verlag, ISBN: 978-3-642-20489-0, 2011.
24. Paluszek, M., and Thomas, S., *MATLAB Machine Learning*, Springer Science+Business Media New York, ISBN: 978-1-4842-2249-2, 2017.
25. Michael Paluszek, Stephanie Thomas, Eric Ham, *Practical MATLAB Deep Learning: A Projects-Based Approach*, Apress, ISBN: 978-1-4842-7911-3, 2022.
26. Kamal I. M. Al-Malah, *Machine and Deep Learning Using MATLAB: Algorithms and Tools for Scientists and Engineers*, Wiley, ISBN: 978-1-3942-0910-1, 2023.
27. Giuseppe Ciaburro, *MATLAB for Machine Learning: Unlock the power of deep learning for swift and enhanced results*, Packt Publishing Pvt. Ltd., ISBN: 978-1-8350-8769-5, 2024.
28. Oliveira, S. and Stewart, D. E., *Writing Scientific Software: A Guide to Good Style*, Cambridge University Press, ISBN: 978-0-521-85896-0, 2006.

COMENTARIOS: Notas de clase y m-files relativas a parte del material a cubrir en el curso serán provistas por los profesores responsables.

Sergio Preidikman
Emmanuel Beltramo

30 de julio de 2024