UNIVERSIDAD NACIONAL DE LA MATANZA

Optimización de la enseñanza de la ciencia matemática aplicada a la economía

Integrantes: Claudio J. Gimeno
Jorge D. Barreto
Alberto Hrynkiewicz
Aníbal Kaech

Director del proyecto: Raimundo Sillitti

3030

ABSTRACT

Institución: Universidad Nacional de la Matanza

Unidad ejecutora: Departamento de Ciencias Económicas

Título del Proyecto: Optimización de la enseñanza de la ciencia matemática aplicada a la economía.

Palabras clave: Optimización – Metodología – Software - Matemática Aplicada – Herramienta

Disciplinas científicas: Matemática Aplicada – Didáctica de la Matemática – Informática aplicada.

Docente Coordinador: Lic. Raimundo Sillitti

Autores: Profesores: Claudio J. Gimeno, Jorge D. Barreto, Aníbal Kaech, Alberto Hrynkiewicz.

Objetivo General: El objetivo general del proyecto consiste en desarrollar estrategias metodológicas necesarias para la optimización de la enseñanza de la ciencia matemática, buscando modelos flexibles a los cambios y utilizando la informática como herramienta.

Marco Teórico: Este trabajo se fundamenta en el concepto de calidad en el ámbito de la educación y en la transmisión de procesos propios de la Matemática más que en la mera transferencia de contenidos. Siendo el medio la herramienta informática, tratando de lograr la familiarización de docente y alumno con software que los asistan y que potencie el trabajo de ambos.

Metodología: El diseño de esta investigación es exploratorio-descriptivo.

Resultados esperados: con el proyecto pretendemos generar material (guías didácticas asistidas por medios informáticos) para mejorar el proceso de enseñanza - aprendizaje de la matemática a nivel superior.

Campos de aplicación: Sistema universitario y superior en general, Universidad Nacional de La Matanza en particular.

PRESENTACIÓN DE LA PROBLEMÁTICA

Numerosos cambios han tenido lugar en años recientes que han afectado profundamente la enseñanza de la matemática en el nivel superior. Entre otros, podemos citar cinco cambios que aún tienen considerable influencia, ellos son:

1. El incremento del número de estudiantes que actualmente cursan estudios superiores.
2. Importantes cambios pedagógicos y curriculares en el nivel pre-universitario.
3. Las crecientes diferencias entre la educación matemática en el nivel medio y la de nivel superior, con respecto a sus propósitos, objetivos, métodos y enfoques de enseñanza.
4. El rápido desarrollo de la tecnología.
5. Presiones sobre las universidades e institutos superiores para que den cuenta públicamente de sus acciones.

Por supuesto todos estos cambios son generales y han tenido influencia en otras disciplinas. Sin embargo, dada su posición central en la educación general, y su naturaleza obligatoria para muchos estudiantes, puede argüirse que estos cambios quizás han tenido una mayor influencia en matemática que en cualquier otra disciplina.

No hay duda que, en muchos países, un número significativamente mayor (respecto a diez años atrás) de estudiantes, están accediendo al nivel superior y tomando cursos de matemática. Por otra parte un porcentaje cada vez menor de estudiantes parece optar por estudios que requieren una cantidad sustancial de matemática. Así, los departamentos universitarios se encuentran frente a un doble desafío. Por un lado, tienen que tratar con el ingreso de estudiantes cuya preparación, conocimientos previos y aún actitudes, son muy diferentes de la que tenían los estudiantes anteriores. Por otra parte, tienen que atraer estudiantes para seguir estudios en matemática, donde las oportunidades de empleo y trabajos bien pagos no parecen ser tan ciertos como en otras disciplinas.

También percibimos muchas veces una discontinuidad entre la educación matemática en el nivel medio y la educación matemática en el nivel superior. Ciertamente, las expectativas y exigencias puestas sobre los estudiantes se incrementan en el nivel terciario.

No se presta la misma atención a las teorías de enseñanza en el nivel universitario como se lo hace en niveles más bajos. Los métodos de enseñanza universitarios tienden a ser más conservadores.

Muchas veces los docentes universitarios tienen responsabilidad simultánea sobre investigación y docencia. Esto es claramente beneficioso pero puede producir que un mayor énfasis sea puesto en investigación matemática en los lugares en que éste es el principal criterio para la promoción en los cargos.

Los profesionales de cursos universitarios de matemática, en general, no han sido entrenados para considerar (y pocas veces consideran) criterios educacionales, didácticos y pedagógicos más allá de la determinación de los contenidos curriculares.

Mundialmente se está haciendo mayor uso de computadoras y calculadoras en la instrucción matemática. Hay muchos software y paquetes de enseñanza disponibles para un gran rango de tópicos curriculares. Esto, por supuesto, plantea la cuestión de qué es lo que estos software y paquetes ofrecen para la enseñanza y aprendizaje del tema, y qué problemas potenciales pueden generar para la comprensión y el razonamiento.

Nuestro proyecto considera que resulta beneficioso juntar aquellos ejemplos donde el uso de tecnología informática y software resultan enriquecedores para la experiencia de los estudiantes y devienen en una mejor comprensión y aprendizaje.

ESTADO ACTUAL DEL CONOCIMIENTO

El origen del cálculo simbólico, entendiendo como tal la manipulación por parte de los programas de expresiones algebraicas no numéricas generando soluciones en modo exacto, comenzó a realizarse a mediados de los años cincuenta. Los trabajos sobre derivación analítica mediante computador de Kahrimanian y Nolan en 1953 pueden considerase como el origen de una nueva disciplina conocida como cálculo simbólico, cálculo formal o álgebra computacional. En los años sesenta comienzan a surgir los primeros sistemas de cálculo simbólico que pueden clasificarse en dos tipos de sistemas:

Sistemas especializados: tratan temas específicos en el campo de la matemática aplicada, se caracterizan por ser dependientes de plataforma, no interactivos.

MACAULAY (especializado en anillos)
LIE (álgebra lineal)
CoCoA (ideales de polinomios)
GAP (grupos)
SHEEP (relatividad).

Sistemas generales: se caracterizan por contener un gran número de funciones y procedimientos predefinidos, son validos en una amplia gama de plataformas y permiten un uso interactivo. Son estos programas los que con el tiempo han ido evolucionando y que en los últimos quince años han revolucionado la enseñanza de las matemáticas y de las ciencias que aplican las matemáticas.

El cálculo simbólico como disciplina científica se divide en las siguientes aplicaciones:

1. Desarrollos en el campo docente y de la investigación en el estudio de ciencias matemáticas, físico-químicas, ciencias de la vida, biología, medicina, farmacia, ciencias sociales, ciencias jurídicas.
2. Desarrollos en investigación y procesos industriales por ejemplo: la pista de atletismo de los últimos juegos olímpicos ha sido diseñada mediante Mathematica.
3. Algebra computacional aplicada a la inteligencia artificial.
4. Algebra computacional "pura".

Posiblemente la característica principal del cálculo simbólico es su carácter interdisciplinar, que permite a profesionales de distintos campos trabajar en un mismo proyecto, con un mismo lenguaje.

Desde el punto de vista de los lenguajes de programación, los programas de cálculo simbólico significan una nueva generación , si la evolución de los lenguajes de programación, desde el código maquina, ensamblador, derivo en los lenguajes de alto nivel, Basic, Pascal, C como un interface entre el lenguaje humano y el código maquina, los lenguajes de programación basados en cálculo simbólico son el interface natural entre el código máquina y el lenguaje de las matemáticas.

Desde el punto de vista docente, y como muestra del impacto que se prevé en la enseñanza mediante el uso de programas de cálculo simbólico, se puede hablar de un nuevo paradigma de enseñanza basado en el cálculo simbólico.

Nosotros hemos desarrollado esta investigación, cuyo objetivo es introducir a los alumnos en el uso del calculo simbólico mediante herramientas informáticas, desde una aproximación interdisciplinar. La motivación de analizar estos programas es doble, por un lado la experiencia acumulada en el uso y desarrollo de productos en estos programas por miembros del departamento y por otro que son muestras significativas de los diferentes programas de cálculo simbólico que existen en el mercado.

Las tareas educativas asistidas por medios informáticos constituyen una idea fuerza, motora de nuevas líneas de pensamiento. Su inserción en el sistema educativo en todos sus niveles, así como la educación continua y a distancia, ha constituido una de las acciones más trascendentes de innovación educativa contemporánea.

En nuestro país es manifiesta la preocupación de la sociedad en general y de la comunidad educativa en particular, por superar ciertas limitaciones que evidencian el modelo de enseñanza - aprendizaje tradicional.

a) El alumno mantiene una actitud pasiva, es simple receptor del contenido.
b) Los tiempos que ocupan los procesos de enseñanza aprendizaje son excesivos.
c) Los alumnos egresan sin estar entrenados en la resolución de problemas y toma de decisiones.
d) Los alumnos fracasan, frecuentemente, en el estudio de las ciencias.

Algunas causas son:

i) La falta de objetivación y conceptualización.
ii) La falta de estrategias para resolver problemas.
iii) Pobre base matemática y de interpretación gráfica.
iv) Escasa base para la percepción de la modelización.

El empleo de los medios informáticos en la enseñanza ayuda a paliar algunos de estos inconvenientes y transforma el modelo de enseñanza – aprendizaje.

En general potencia facultades que no han sido desarrolladas en el aprendizaje tradicional.

Es sabido que en otros países ya se han logrado avances concretos hacia un nuevo modelo de enseñanza – aprendizaje, numerosas Universidades extranjeras conducen programas de desarrollo de guiones didácticos. La Universidad de Londres, con sus tres Colegios (Chelsea, Queen Mary e Imperial),ha liderado un Plan Nacional de Desarrollo del Aprendizaje Asistido por Computadora, hoy vigente en mas de 600 Universidades y Colegios Ingleses.

Debe agregarse como uno de los Proyectos Europeos mas importantes el del Ministerio de Educación de Francia y de las Universidades de París VI y VII, así como el del Ministerio de Educación de España.

En los Estados Unidos de América se distinguen como centros importantes de esta especialidad, el Educational Technology Center, Universidad de California, Irvine; el consorcio computacional Educativo de la Universidad de Minnesota;el proyecto de la Universidad de Stanford, y el proyecto del Technical Education Research (TERC) de Massachusetts, entre otros.

Consideramos que la Argentina no puede ignorarlos, y debe, organizarse para recorrer ese camino.

Las instituciones patrocinantes y las universidades deben considerar que las tareas educativas asistidas por medios informáticos son una nueva rama multidisciplinarias del saber, y que el desarrollo de su actividad constituye una tarea de investigación similar a otras ramas de la ciencia.

La realización de guías didácticas debe considerarse como una obra de investigación y evaluarse como tal, pues significa un acto de creación realizado con metodología científica y dedicación completa de un grupo de estudiosos que se encuentran en la frontera del conocimiento.

Numerosas universidades extranjeras conducen programas de desarrollo de guías de tareas didácticas asistidas por medios informáticos, preponderantemente en las ciencias exactas y naturales.

En la integración de la informática con la educación debe primar la concepción pedagógica sobre el aspecto técnico. Para asegurar esto es necesario que:

- Las guías sean producidas por requerimiento del sistema de enseñanza y subordinados al mismo. Deben producirse para enriquecer el proceso de aprendizaje previamente establecido.
- Las guías deben ser evaluadas dentro del sistema de enseñanza diseñado y desarrollado.

Dado el estado actual de la tecnología informática, de la tecnología educativa y de la ciencia cognitiva, se requiere que las guías didácticas se desarrollen por un grupo integrado de docentes, de acuerdo con últimos avances científicos, a fin de mantener el nivel académico y la actualización en lo que hace al contenido.

PROBLEMA

¿La complejidad del cálculo complementario desvía la atención del alumnos y deja poco espacio para la discusión y análisis del problema planteado?

HIPÓTESIS

El uso del software adecuado en la enseñanza de la Matemática, permite que el alumno se concentre en el problema planteado, porque evita que el cálculo secundario desplace en importancia el problema a resolver.

OBJETIVOS

El objetivo general del proyecto consiste en desarrollar estrategias metodológicas necesarias para la optimización de la enseñanza de la ciencia matemática, buscando modelos flexibles a los cambios y utilizando la informática como herramienta.

OBJETIVOS SECUNDARIOS

Analizar y comparar diferentes software de aplicación a la matemática, buscando posibles ventajas y desventajas de su utilización.

Seleccionar los software más adecuados en función a los contenidos procedimentales de las asignaturas: Matemática I y Matemática II del Departamento de Ciencias Económicas de la Universidad Nacional de La Matanza, para elaborar guías didácticas.

METODOLOGÍA

Existe una tendencia internacional en producir un cambio en la presentación de la enseñanza de las ciencias, mediante la utilización de todos los medios disponibles (texto, medios informáticos, gabinetes).

Es dentro de este marco de referencia donde hemos ubicado nuestro proyecto.

Como docentes aspiramos que el alumno adquiera la habilidad para practicar la metodología científica.

Particularmente en lo que respecta al uso del software, que adquiera la habilidad para objetivar e interpretar los resultados.

La generación de un material adecuado es un delicado problema. Un material que no tiene un buen enfoque pedagógico puede finalmente, perjudicar al alumno.

Guía didáctica y un software adecuado son los portadores del mensaje instruccional para conducir el aprendizaje. Supone una serie de pasos para el relevamiento y tal vez el diseño y producción de los mismos.

Los pasos metodológicos que guiaron nuestro proyecto fueron:

a) Identificación de las necesidades de los alumnos.
b) Planificación del contenido.
c) Fijación de objetivos y metas.
d) Relevamiento del software actual adecuado para esos objetivos o metas.
e) Especificación del criterio pedagógico que lo asiste.
f) Realización de pruebas pilotos en distintos momentos y niveles.
g) Evaluación formativa.
h) Revisión del software seleccionado y el material teórico nuevamente sobre la base de la evaluación formativa.
i) Determinación de efectividad (o sea, determinar con que nivel de efectividad se ha adquirido el aprendizaje).
j) Revisión de los déficits y recomendaciones de modificaciones permanentes.
k) Confección de guías didácticas que compendien los anteriores items.
l) Divulgación de las experiencias y los resultados.
ll) Estudio de la metodología de difusión.

PLAN DE TAREA

Identificadas las dificultades tradicionales en la enseñanza de las ciencias exactas y observando que falta una herramienta que ayude a manipular información rápidamente, que permita confeccionar gráficos con facilidad, a cambiar situaciones y a obtener nuevos gráficos, en forma instantánea, definimos los objetivos de nuestra investigación.

En el desarrollo de la misma y en busca del software más adecuado para llevar adelante un guión didáctico, en algunas cuestiones temáticas particulares, relevamos los siguientes programas:
MATHEMATICA (versión 3.0), DERIVE (para Windows 95), MATHCAD (plus 6.0), FW (funciones para Windows), MATLAB (5.0), MATHEASS (única versión).

Analizamos en cada uno de ellos las siguientes variables:

- Nivel de dificultad de su sintaxis.
- Manejo previo de un sistema operativo
- Idioma de la versión disponible
- Claridad de visualización (interfase)
- Definición en la confección de gráficos.
- Campo de aplicación.
- En que medida fueron relevados(trabajos previos sobre el software

Y con el estudio citado confeccionamos la siguiente tabla:

Software	Sintxis	Manejo Previo	Idioma	Interfase	Definición Gráfica	Nivel de estudio	Campo de aplicación
DERIVE	Medio	Dos Windows 95	Español	Buena	Adecuada	Muy analizado	Amplia
MATHCAD	Medio	Windows 95	Inglés	Muy buena	Adecuada	Muy analizado	Amplia
FW	Bajo	Windows 95	Español	Muy buena	Óptima	Median. Analizado	Básica
MATLAB	Medio	Windows 95	Español	Regular	Regular	Median. Analizado	Técnica
MATHEMATICA	Alto	Windows 95	Inglés	Muy bueno	Óptima	Poco Analizado	Amplia
MATHEASS	Medio	DOS	Alemán	Regular	------------	Poco Analizado	Escaso

Como datos relevantes, tanto el DERIVE como el MATHCAD resultaron dos aplicaciones ampliamente desarrolladas y analizadas. Ambas versiones (para Windows) permiten un versátil manejo y claridad de interfase. La dificultad principal en el DERIVE tradicional es que se requiere conocimientos básicos de manejo del sistema operativo DOS, esto no sucede con la versión para Windows 95, pero el inconveniente es que no está lo suficientemente difundida.. En tanto el MATHCAD presenta la dificultad del idioma (la versión 6.0 está en inglés) lo que implica que aquel que para operarlo se necesita estar en una etapa de la carrera que garantice algún nivel de Ingles ya cursado.

El software MATHEASS presenta la dificultad del idioma (la versión difundida está en alemán) si bien es de fácil manejo y de sencilla comprensión requiere a raíz del idioma en que está desarrollado un conocimiento previo de los contenidos a trabajar.

Los tres sofware MATHEMATICA, MATLAB y FW, fueron sometidos a un análisis más profundo ya que el relevamiento inicial nos permitió determinar que se encontraban con un potencial de aplicación superior al resto respecto del objetivo que nos guiaba.

Hemos desarrollado los siguientes temas:

- Integrales definidas
- Integral indefinida
- Sistema de inecuaciones
- Serie de Fourier
- Recta de regresión
- Derivadas
- Extremos condicionados
- Estudio de funciones
- Número complejos
- Matrices y vectores
- Determinantes
- Trigonometría
- Gráfico de funciones

Finalizado este primer relevamiento centramos nuestra atención en tres de estos programas.

En particular el software FW (funciones de Windows) es el que presenta mayor facilidad en su manejo y resolución para una mayor cantidad de temas programáticos, a saber:

- Estudio de funciones
- Integral definida
- Cálculo de áreas
- Serie de Fourier
- Recta de regresión

Este software permite ser utilizado con requisitos mínimos de computación, está en idioma español y tiene una importante resolución gráfica, por todo las guías fueron diseñadas con el propósito de ser directamente implementadas en la cátedra de Matemática I del Departamento de Ciencias Económicas de la Universidad Nacional de La Matanza.

En tanto los software MATHEMATICA y MATLAB presentan una sintaxis más compleja a la vez que alcanzan un mayor grado de potencialidad desde el punto de vista programático curricular. Por tal motivo consideramos que sería conveniente para su utilización una primera aproximación del alumno al manejo de herramientas informáticas a través de un software más básico (FW) como el desarrollado en la presente investigación.

CONCLUSIONES

La dificultad intrínseca que presenta la enseñanza de la matemática es la manipulación de datos obtenidos, su representación gráfica en función de diferentes parámetros y su posterior análisis. Este primera parte (previa al análisis) es un trabajo tedioso por las dificultades presentadas y el consumo de tiempo. Así, la discusión de los resultados, queda relegada a último término y en la mayoría de los casos no se complimenta.

Consideramos que la aplicación de un programa de estudio que estimule la utilización de herramientas informáticas encausadas a través de guías didácticas constituyen una manera importante de empezar a comprender y disfrutar el aprendizaje de esta asignatura.

Como uno de los resultados de la investigación diseñamos guías didácticas referidas a temas de la asignatura Matemática I, Departamento de Ciencias Económicas, de la Universidad Nacional de la Matanza.

Los temas seleccionados, lo fueron en función de que permitían optimizar las variables en estudio, en cada uno de los software relevados.

Creemos que una importante razón por la cual a nuestros estudiantes no les gusta ni les interesa la Matemática es porque no le encuentran relación alguna con el mundo cotidiano, con sus problemas personales ni con su futuro profesional.

En este sentido la informática se encuentra ya entre nosotros, en lo personal y en lo profesional, por ese motivo consideramos que la informática no debe estar alejada de la enseñanza de la matemática, pero cumpliendo el rol anteriormente indicado, actuar como herramienta que ayude al alumno a conceptualizar, razonar, generalizar y cuestionar e investigar.

Lo hasta aquí realizado podría aportar una mejora significativa (cuantificable) en la calidad y cantidad de contenidos implementados en la cátedra de Matemática I.

Sería interesante (por no decir necesario) continuar esta metodología en el segundo curso de matemática para los estudiantes de ciencias económicas de esta casa de altos estudios.

BIBLIOGRAFÍA

BARRÓN RUÍZ A. Aprendizaje por descubrimiento: principios y aplicaciones inadecuadas. U. de Barcelona. 1996.

BIRRKHOFF G. A Survey of Modem Algebra Ed. Macmillan. N.Y. 1953.

BLUM E. Numerical Analysis and Computations: Theory and Practice Ed. Adisson – Wesley. 1972.

BURDER R. Numerical Analysis Ed. Mc. Graw-Hill. N.Y. 1972.

CARRILLO L. Derive: Aplicaciones Matemáticas para PC Ed. Ra-ma. 1994.

CONTE S. Elementary Numerical Analysis Ed. Mc. Graw-Hill. N.Y. 1972.

DIEUDONNÉ J. Fundamento de Análisis Moderno Ed. Reverté. Barcelona. 1966.

GONZÁLEZ C. Fundamentos de la Optimización de la Matemática con Derive y Mathematica Ed. Ra-Ma. 1997

HILL W. Teorías Contemporáneas del Aprendizaje Ed. Paidós. México. 1990.

LIPSCHUTZ S. Álgebra Lineal Ed. Mc. Graw-Hill. España. 1966.

NAKAMURA T. Análisis Numérico y Visualización Gráfica con Matlab Ed. Prentice Hall. N.Y. 1997.

PAULOGORRÁN J. Cálculo Matemático con Derive para PC Ed. Ra-Ma. 1997

PÉREZ C. Cálculo Simbólico y Numérico con Mathematica Ed. Ra-Ma. 1995.

REY PASTOR J. Lecciones de Álgebra Madrid. 1930.

REY PASTOR J. Elementos de Análisis Algebraico Madrid. 1930.

SCHOENFELD A. Ideas y Tendencias en la Resolución de Problemas. O.M.A.

SOTO M. Álgebra Lineal con Matlab y Maple Ed. Adison-Wesley Iberoamericana. España. 1995

FICHA 1

SOFTWARE: FW (funciones para Windows)

TEMA: Integrales Definidas

Matemática I

Tema:

Integral

Conocimientos previos:

Áreas. Derivación. Cálculo de primitivas.

Nivel :

Primer curso universitario de carreras de Contador Público y Licenciado en Administración

Inserción Curricular:

Cálculo Diferencial e Integral.

Objetivos:

Conocer los conceptos de: Integral definida. Función área.

Darse cuenta de que las funciones área son funciones primitivas. Teorema fundamental del cálculo.

Comprender, y saber aplicar, el "Segundo teorema fundamental del cálculo", regla de Barrow.

Procedimiento:

Integral definida

Ejecutar el programa: **Funciones para Windows**

Representemos la siguiente función: **F(x)=1/20(x^3+2x^2-11x+38)** . Le llamaremos función**1**

Valores de los ejes:

Origen eje X	-7.5
Unidad eje X	1
Final eje X	7.5
Origen eje Y	-4
Unidad eje Y	1
Final eje X	6

Calcular la integral definida entre -4 y 1.

Menú **1 f.** *.Opción* **Integral definida....** *Escribir como extremos de integración los dados.*

- Mediante el portapapeles dibujar la función en este lugar.

- Escribir el valor: ______________________.

Lo que representa: **Área limitada por la curva, eje de abscisas, rectas: x=-4 y x=1**

- Calcular la integral definida entre -4 y -1: ______________________.

- Calcular la integral definida entre -1 y 1: ______________________.

- Suma los resultados: ______________________.

Observa la siguiente propiedad:

(Int. def. entre -4 y -1) + (Int. def. entre -1 y 1) = (int. def. entre -4 y 1)

Función área

Calcular una primitiva de la función anterior: **1/20*(x^4/4+2/3x^3-11/2x^2+38x)**

Ejecutar de nuevo programa **Funciones para Windows**. Simultáneamente con el anterior.

Representemos la función primitiva: **F(x)=1/20*(x^4/4+2/3x^3-11/2x^2+38x)**

Calcular la imagen de -4: _______________________.

Representemos la función primitiva:
F(x)=1/20*(x^4/4+2/3x^3-11/2x^2+38x)+10.933. Le llamaremos función **2** (Hemos escogido una primitiva en la cual la imagen del -4 vale 0).

Valores de los ejes:

Origen eje X	-7.5
Unidad eje X	1
Final eje X	7.5
Origen eje Y	-4.4
Unidad eje Y	1.4
Final eje X	6.4

- *Calcular la imagen de -4:*_______________________.

- *Calcular la imagen de -1:*_______________________.

- *Calcular la imagen de 1:* _______________________.

- *Observar la siguiente propiedad:*

Las imágenes de x de la función 2 son las integrales definidas entre -4 y x de la función 1.

Anteriormente hemos visto que la integral definida era el área que quedaba debajo de la curva. Esta nueva función (función 2) mide esta área. Le llamamos:
Función área

La función que utilizamos como función área (función 2) es una primitiva de la función 1.
Se puede demostrar que esto ocurre siempre. Es decir:

Dada una función, si tiene funciones primitivas, éstas son funciones área.

Esto se denomina: **Teorema fundamental del cálculo.**
(También se enuncia: Dada una función, si tiene una función área, la derivada de la función área es la función original).

Cálculo de la integral definida entre dos puntos mediante una función área

Para calcular la integral definida entre, por ejemplo, -1 y 1, podemos hacerlo mediante la función área (función 2). La imagen de 1 es el área entre -4 y 1. La imagen de -1 es el área entre -4 y -1. (ver apartado 2.2).

- Restamos estos dos valores: ________________________.

Que es el valor encontrado en el apartado 2.1, de la integral definida entre -1 y 1 (empleando la función 1)

Esta forma de hallar la integral definida entre 2 valores, restando las imágenes de una función primitiva, se conoce como: **Segundo teorema fundamental del cálculo** o **regla de Barrow**.

Distintas funciones área

Hemos visto que la función área (función 2) es una primitiva de la primera función (función 1). Pero primitivas de una función hay muchas. Si cambiamos la constante 10.933 por cualquier número, la función resultante también cumple el hecho de ser una primitiva de la función 1. Lo que ahora comprobaremos es que, si utilizamos otra primitiva, podemos seguir calculando la integral definida entre 2 puntos restando sus imágenes, calculadas en la nueva función.

- Representemos una nueva función primitiva:
F(x)=1/20*(x^3/4+2/3x^3-11/2x^2+38x). Le llamaremos función **3**.

Valores de los ejes:

Origen eje X	-7.5
Unidad eje X	1
Final eje X	7.5
Origen eje Y	-4.4
Unidad eje Y	1.4
Final eje X	6.4

- *Calcular la imagen de -1:*_______________________.

- *Calcular la imagen de 1:* _______________________.

- *Restar estos dos valores:* _______________________.

Y observamos el mismo resultado. Es decir: para calcular la integral definida entre 2 puntos de una función de la cual conozcamos una primitiva (Para ello se estudia el **cálculo de integrales indefinidas**), se calcula la diferencia de sus imágenes. El resultado no depende de la primitiva escogida.
De hecho las distintas primitivas sólo se diferencian en una constante aditiva. Lo que hace que, al restar 2 imágenes, esta constante se cancele.

Para ver la forma de las distintas primitivas o funciones área, representaremos 4 al mismo tiempo.

Representar:
F(x)=1/20*(x^3/4+2/3x^3-11/2x^2+38x)
G(x)=1/20*(x^3/4+2/3x^3-11/2x^2+38x)+4
H(x)=1/20*(x^3/4+2/3x^3-11/2x^2+38x)-8
I(x)=1/20*(x^3/4+2/3x^3-11/2x^2+38x)+16

- *Mediante el portapapeles dibujar la función en este lugar.*

Observar que tienen la misma forma. Sólo se distinguen por una traslación vertical.

FICHA 2

SOFTWARE: FW (funciones para Windows)

TEMA: Estudio de Funciones

Matemática I

Tema:

Estudio general de una función.

Nivel :

Primer año de la carrera de Contador Público y Licenciado en Administración.

Conocimientos previos:

Resolución de ecuaciones. Cálculo de derivadas.

Objetivo general:

Comprender y saber aplicar las distintas técnicas que ofrece el cálculo diferencial para el estudio y representación de funciones.

Objetivos específicos:

Comprender y saber calcular los siguientes conceptos:

- Raíces.
- Ordenada en el origen.
- Máximos relativos.
- Mínimos relativos.
- Intervalos de crecimiento.
- Intervalos de decrecimiento.
- Puntos de inflexión.
- Intervalos de concavidad.
- Intervalos de convexidad.
- Asíntota vertical.
- Asíntota horizontal.
- Asíntota oblicua.
- Simetría: par, impar.

Procedimiento:

Ejecutar el programa: **Funciones para Windows**

Condiciones de trabajo:

- Ventana maximizado. *Pulsar en la ventana el botón de maximización, esquina superior derecha.*

- La opción "**Baja precisión**" del menú "**Opciones**" **activa**. Creemos que es mejor así. Recordar que, por defecto, al arrancar el programa aparece ya activa. Como esta opción sólo afecta a los puntos singulares, máximos, mínimos, intervalos..., puede provocar ligeras diferencias con el cálculo de imágenes. Si quiere evitar estas diferencias, desactive esta opción.

- La opción "**Trazar cálculos**", del mismo menú, creemos que debe estar **activa** (opción por defecto), ya que ayuda mucho a la comprensión de los conceptos. Si la computadora no es muy rápida se aconseja desactivarla.

Raíces, ordenada en el origen.

Representemos la siguiente función: **F(x)=1/36(3x^4-20x^3+12x^2+96x-110)** .
La llamaremos función **1**.

Valores de los ejes:
Origen eje X
-6.5
Unidad eje X
1
Final eje X
8.5
Origen eje Y
-5
Unidad eje Y
1
Final eje X
5

Calcular las raíces.

Menú 1 **f.** *.Opción* **Raíces.**

- Mediante el portapapeles dibujar la función en este lugar.

*- Escribe las raíces*_______________.

Las raíces son **las intersecciones de la función con el eje de abcisas.**
Calcular la ordenada en el origen.

Menú 1 **f.** *.Opción* **Imagen....**

- Calcular la imagen de 0: _______________.

La ordenada en el origen es **la intersección de la función con el eje de ordenadas**. O lo que es lo mismo, la imagen de 0.

Máximos y mínimos relativos.

Calcular los máximos.

Menú 1 **f.** *.Opción* **Máximos.**

- Escribe los máximos: _______________.

Calcular los mínimos.

Menú 1 **f.** .Opción **Mínimos.**

- Mediante el portapapeles dibujar la función en este lugar.

- Escribe los mínimos: _______________.

Vamos a ver con más detalle el significado de estos conceptos. Concretaremos en el mínimo (-4,-1.28). Para ello, procedamos del siguiente modo:

Menú 1 **f.** *.Opción* **Imagen....**

*- Calcular la imagen de -4:*_______________.

Pulsar el botón **d->**

La imagen de **4.025** que es _______________.

Pulsar dos veces el botón **<-i**

Obtenemos la imagen de **3.975**, es _______________.

Observar que sus imágenes son mayores que la imagen de -4, _______________. Si calculamos las imágenes de los puntos alrededor de -4, vemos que todas son mayores. Si nos desplazamos mucho, por ejemplo, buscando la que es imagen del 0 que es _________________________. vemos que no. Este es el concepto de **mínimo relativo**. En un punto **a** (en nuestro caso -4) del eje de abcisas diremos que existe un mínimo relativo para la función **F(x)**, si, existe un entorno de a en el cual las imágenes de los valores distintos de **a**, son mayores que la imagen de **a**.

Intervalos de crecimiento y de decrecimiento.

Calcular los intervalos de crecimiento.

Menú **1 f.** *.Opción* **Intervalos de crecimiento**.

- Mediante el portapapeles dibujar la función en este lugar.

- Escribir los intervalos: _______________.

¿Qué significan?. Si observamos la gráfica de izquierda a derecha, vemos que son precisamente la parte de la gráfica con sentido ascendente.

Precisemos un poco más ésto: veamos qué ocurre en un punto que pertenezca a uno de los dos intervalos, por ejemplo, el 5.5.

Menú **1 f.** *.Opción* **Imagen....**

*- Calcular la imagen de -5.5:*_______________.

Pulsar el botón **d->**

La imagen de **5.525** que es _______________.

Pulsar dos veces el botón **<-i**

Obtenemos que la imagen de **5.975** es ________________.

Observamos que la imagen a su derecha (del 5.5) es mayor y menor a su izquierda. Por ello definimos que en un punto la función es creciente si existe un entorno de este punto, en nuestro caso 5.5, en el cual las imágenes a su derecha son _____. y _____.a su izquierda.

Calcular los intervalos de decrecimiento.

Menú 1 **f.** *.Opción* **Intervalos de decrecimiento**.

- Escribir los intervalos: ________________.

¿Qué significan?. Si observamos la gráfica de izquierda a derecha, vemos que son precisamente la parte de la gráfica con sentido descendente.

Precisemos un poco más ésto. Vamos a ver que ocurre en un punto que pertenezca a uno de los dos intervalos, por ejemplo, el 3.5.

Menú 1 **f.** *.Opción* **Imagen....**

Calcular la imagen de -3.5: ________________.

Pulsar el botón **d->**

La imagen de **3.525**, es ________________.

Pulsar dos veces el botón **<-i**

Obtenemos que la imagen de **3.475** es ________________.

Observamos que la imagen a su derecha (del 3.5) es menor y mayor a su izquierda (recordar que son números negativos). Por ello definimos que en un punto la función es decreciente si existe un entorno de este punto, en nuestro caso 3.5, en el cual las imágenes a su derecha son menores y mayores a su izquierda.

Pongamos orden:

_______________. **decreciente.** _______________. **creciente.**
_______________. **decreciente.** _______________.**creciente.**

Relación entre los anteriores conceptos y la derivada.

Hemos aprendido los conceptos de máximo, mínimo, intervalos de crecimiento y decrecimiento. Ahora veremos un método para hallarlos utilizando la derivada.

Relación entre intervalos de crecimiento y decrecimiento y la derivada.

Calculemos la derivada en un punto donde la función es decreciente, 3.6.

Menú **1 f.** *.Opción* **Derivada en un punto....**

Calcular la derivada de -3.6: _______________.

Calcular la derivada en los puntos contiguos mediante los botones **<-i** y **d->**.

Observamos que siempre son valores con signo _______________.

Ahora calcularemos la derivada en un punto donde sea creciente.

Calcular la derivada de -4.5: _______________.

Calcular la derivada en los puntos contiguos mediante los botones **<-i** *y* **d->**.

Observamos que siempre son valores con signo _______________.

Con todo ello podemos concluir:
- Una función es decreciente en un punto cuando su derivada es _____________.
- Una función es creciente en un punto cuando su derivada es _______________.

Veamos qué ocurre en los puntos donde la derivada no es ni positiva ni negativa. Es decir, cuyo valor sea 0.

Menú **1 f.** *.Opción* **Derivada en un punto....**

Calcular la derivada de 3.8: _______________.

Calcular la derivada en los puntos a su derecha mediante el botón **d->**. *Hasta que el valor sea positivo. Observar la forma de la recta tangente representada.*

Observamos que cuando x=4, la recta tangente es horizontal y su pendiente (la derivada) vale: _______________. Recordar que en el 4 hay un ____________.

Calcular la derivada de 1.9: _______________.

Calcular la derivada en los puntos a su derecha mediante el botón **d->**. *Hasta que el valor sea negativo. Observar la forma de la recta tangente representada.*

Observamos que, cuando x=2, la recta tangente es horizontal y su pendiente (la derivada) vale: _______________. Recordar que en el 2 hay un ___________.

De todo ello podemos sacar la siguiente conclusión:

En los máximos y mínimos relativos de una función la derivada se _______.

También hemos visto cómo diferenciar un máximo de un mínimo. En el primer caso, x=4, mínimo, la derivada era negativa para números menores que 4. Se hacía 0 en el 4. Y positiva para números mayores que 4. Es decir, la función derivada es creciente en el 4. Pero hemos visto que una función es creciente cuando su derivada es positiva. A consecuencia de ésto:

En un punto hay un **mínimo**, si la **derivada =** _________________. y la **segunda derivada es** ________________. (Así la primera derivada es creciente).

En el segundo caso, x=2, máximo, la derivada era positiva para números menores que 2. 0 en el 2. Negativa para números mayores que 2. Es decir, la función derivada es creciente en el 2. Pero hemos visto que una función es decreciente cuando su derivada es negativa. Lo que implica que:

En un punto hay un **máximo**, si la **derivada =** _________________. y la **segunda derivada es** ________________. (Así la primera derivada es decreciente).

Para ver un poco mejor ésto, dibujemos la función derivada.

Menú **1 f.** *.Opción* **Función derivada**.

Observamos que los puntos donde se hallan los máximos y mínimos es donde la función derivada (en verde) corta el eje de abcisas .

En el máximo, la función derivada es decreciente. En los mínimos, decreciente.

Resumen:

Una función es **creciente** en un punto x ,si ________________.

Una función es **decreciente** en un punto x ,si ________________.

Una función tiene un **máximo** en un punto x, si ___________. y ______________.

Una función tiene un **mínimo** en un punto x, si ___________. y ______________.

Puntos de inflexión. Intervalos de concavidad y de convexidad.

Calcular los intervalos de concavidad.

Menú **1 f.** *.Opción* **Intervalos de concavidad**.

- Mediante el portapapeles dibujar la función en este lugar.

- Escribir los intervalos: ________________.

Calcular los intervalos de convexidad.

Menú **1 f.** *.Opción* **Intervalos de convexidad**.

- Escribir los intervalos: ________________.

Podemos describir los dos tipos de intervalos, según vemos en la gráfica, de la siguiente manera.

Valles: ________________.

Montañas: ________________.

Precisaremos mejor estos conceptos cuando los relacionemos con la derivada.

Calcular los puntos de inflexión.

Menú **1 f.** *.Opción* **Intervalos de concavidad**.

-Escribir los puntos de inflexión: ________________.

Dentro de los márgenes de error, vemos que los límites de los intervalos de concavidad y convexidad son los puntos de inflexión.

Así:

Punto de inflexión, son los puntos donde la gráfica cambia de cóncava a convexa o viceversa.

Relación entre los anteriores conceptos y la derivada.

Calculemos la derivada en un punto donde la función es cóncava, -0.8.

Menú **1 f.** *. Opción* **Derivada en un punto....**

-Calcular la derivada de -0.8: ________________.

-Calcular la derivada en los puntos a su derecha mediante el botón **d->**, *hasta, por ejemplo, -0.45:* ________________.

Observamos que la recta tangente queda **debajo** de la curva en todos los casos. También hemos podido comprobar que el valor de la derivada iba aumentando. Es decir, la función derivada es **creciente** en los puntos donde la función es cóncava.

Ahora calcularemos la derivada en un punto donde sea convexa.

- Calcular la derivada de 1.4: ________________.

- Calcular la derivada en los puntos a su derecha mediante el botón **d->**, *hasta, por ejemplo, 1.8:* ________________.

Observamos que la recta tangente queda por **encima** de la curva en todos los casos. También hemos podido comprobar que el valor de la derivada iba disminuyendo. Es decir, la función derivada es **decreciente** en los puntos donde la función es convexa.

Para comprender mejor ésto, proseguiremos con:

Menú **1 f.** *. Opción* **Segunda derivada**.

Observamos lo siguiente:
Una función es cóncava donde la primera derivada es creciente, que equivale a que la segunda derivada sea: ________________.
Cuando sea convexa, la segunda derivada será: ________________.
En los puntos de inflexión la segunda derivada vale: ________________.

Asíntotas verticales.

Representar la siguiente función: **F(x)=1/((x+3)*(x-1))+1** .
Le llamaremos función **2**.

Valores de los ejes, los iniciales.

Calcularemos las imágenes de los puntos cercanos al -3.

Menú **1 f.** .*Opción* **Imagen....**

- *Calcular la imagen de -3.1:* ________________.

Pulsar sucesivamente el botón **d->**.
Vemos que la imagen de -3.025 vale ______. y -3 __________. Si vamos más a la derecha, éstas tienen valores negativos.

Si calculamos imágenes más próximas al -3, por la izquierda.

f(-3.01)= ________________.
f(-3.001)= ________________.
f(-3.0001)= ________________.

Vemos que los valores son cada vez mayores. Cuando las imágenes de una función, al acercarnos a un punto, se hacen cada vez mayores, (tienden a infinito o a menos infinito) decimos que la función tiene una **asíntota vertical**.

Podemos ver que si nos acercamos a -3 por la derecha, las imágenes tienden a menos infinito.

f(-2.99)= ________________.
f(-2.999)= ________________.
f(-2.9999)= ________________.
f(-2.999999)= ________________.

De hecho, se define **asíntota vertical** como la ecuación de una recta vertical, x=a, donde a es el punto singular.

Vemos esas rectas:

Menú **1 f.** .*Opción* **Discontinuidades aisladas.**

- Mediante el portapapeles dibujar la función en este lugar.

Las ecuaciones de las asíntotas de esta función son:

__________________. __________________.

Asíntotas Horizontales.

*Representar la anterior función:***F(x)=1/((x+3)*(x-1))+1**, *juntamente con:* **G(x)=1**. Le llamaremos función **3**.

-Mediante el portapapeles dibujar la función en este lugar.

Observamos que,para los valores extremos de la representación, las dos gráficas tienden a confundirse. Cuando esto ocurre, decimos que la función tiene una **asíntota horizontal**. En nuestro caso _________________.

En general, la ecuación de una recta se expresa: **y=ax+b** donde **a** es la pendiente, que, en nuestro caso, por ser una recta horizontal, siempre vale **0** y donde **b**, la ordenada en el origen (término independiente), se calcula mediante el siguiente límite:

b = lim f(x)
x->infinito

Asíntotas Oblicuas.

Representar las funciones: **F(x)=x^2/(x-2)**. Función **4**. **G(x)=x+2**. Función **5**.

Valores de los ejes:
Origen eje X
-10
Unidad eje X
2
Final eje X
20
Origen eje Y
-10
Unidad eje Y
2
Final eje X
20

- Mediante el portapapeles dibujar la función en este lugar.

Observamos que, para los valores extremos de la representación, las dos gráficas tienden a confundirse. Cuando esto ocurre decimos que la función tiene una **asíntota oblícua**. En nuestro caso, **y=x+2**.

Para hallar la ecuación de esta recta, y=ax+b:

$$a=\lim_{x\to\infty}\frac{f(x)}{x} \quad y \quad b=\lim_{x\to\infty} f(x)-ax$$

$$a=\lim_{x\to\infty}\frac{f(x)}{x} \quad y \quad b=\lim_{x\to\infty} f(x)-ax$$

Simetría: Par, impar

Decimos que una función tiene simetría par, **función par**, cuando **f(x)=f(x-)**.
Decimos que una función tiene simetría impar, **función impar**, cuando **f(x)=-f(x-)**.

Veamos el significado de ésto.

- Representar la siguiente función: **F(x)=1/((x^2-2)**. Función **6**.

- Mediante el portapapeles dibujar la función en este lugar.

*- Calcular las imágenes de:*1, -1; 3.5, -3.5; -2, 2; 5, -5.

F(1)= ________________. F(-1)= ________________.
F(3.5)= ________________. F(-3.5)= ________________.
F(-2)= ________________. F(2)= ________________.
F(5)= ________________. F(-5)= ________________.

Observamos que la imagen de un número es igual a la de su opuesto. Cuando una función cumple esta propiedad decimos que es **par**. También se denomina, **simétrica respecto al eje Y**. El eje Y actúa como un espejo.

- *Representar la siguiente función:* **F(x)=2/(x^3-2x)**. Función **7**.

- *Mediante el portapapeles dibujar la función en este lugar.*

- *Calcular las imágenes de:*1, -1; 3.5, -3.5; -2, 2; 0.5, -0.5.

F(1)= ________________. F(-1)= ________________.

F(3.5)= ________________. F(-3.5)= ________________.

F(-2)= ________________. F(2)= ________________.

F(0.5)= ________________. F(-0.5)= ________________.

Observamos que la imagen de un número es igual a la de su opuesto cambiada de signo. Cuando una función cumple esta propiedad decimos que es **par.** También se denomina, **simétrica respecto al origen de coordenadas**.

Resumen:

Para representar una función, se procede del siguiente modo:

Se calcula:

Dominio.

Cortes con los ejes:

Raíces.

Ordenada en el origen.

Máximos relativos.

Mínimos relativos.

Intervalos de crecimiento.

Intervalos de decrecimiento.

Puntos de inflexión.

Intervalos de concavidad.

Intervalos de convexidad.

Asíntotas verticales.

Asíntotas horizontales.

Asíntotas oblicuas.

Simetrías.

- Para calcularlos, actuamos tal como hemos visto anteriormente.
- Debe tenerse en cuenta que no suele ser necesario calcular todos y cada uno de los puntos.
- Después se traslada todo al dibujo de la gráfica.

Probar con la siguiente gráfica: f(x)=x^3/(x^2-2)

Si escogemos como valores de los ejes:

Origen eje X

-7.5

Unidad eje X

1

Final eje X

7.5

Origen eje Y

-6

Unidad eje Y

1

Final eje X

6

- Mediante el portapapeles dibujar la función en este lugar.

- Imprimir este documento.

FICHA 3

SOFTWARE: FW (funciones para Windows)

TEMA: Recta de regresión

Matemática II

Tema:

Regresión.

Conocimientos previos:

Estadística básica. Cálculo de: medias, desviación típica.

Nivel :

Segundo año de la carrera de contador público y licenciado en administración.

Inserción Curricular:

Optimización

Objetivos:

Conocer el concepto de relación lineal estadística.

Significado del coeficiente de correlación lineal.

Significado de la ecuación de la recta de regresión.

Predicción.

Otras ecuaciones de regresión.

Procedimiento:

1- Planteemos el siguiente problema:

Sean 10 alumnos de matemática II seleccionados al azar. Se nos da los pesos y las alturas en la tabla siguiente:

altura (metros)	Peso (Kg)
1.91	84
1.78	78
1.77	77
1.87	86
1.75	65
1.65	46
1.66	49
1.68	60
1.74	60
1.68	60

Las preguntas que tratamos de contestar son:

1 - ¿Existe alguna relación entre estos datos?

2 - ¿Si existe, en qué medida es cierta?

3 - ¿Cual es la relación?.

4 - ¿Cómo utilizarla para conocer valores no incluidos en la tabla?.

Procedamos de la forma siguiente:

Ejecutar el programa: **Funciones para Windows.**

Pulsar el botón **Función numérica.**

Escoger una función, activando el botón **regresión.**

Introducir los datos de la tabla anterior. Recordar que podemos utilizar el portapapeles.

Pulsar **Aceptar.**

Contestar **Si** *a la modificación de los ejes.*

Pulsar **Aceptar** *en el cuadro Principal.*

- Mediante el portapapeles dibujar la función en este lugar.

Vemos representados los puntos y una recta. De momento sólo nos fijaremos con los puntos:

Observar que hay una cierta tendencia en el sentido de que a valores altos de las alturas corresponde valores mayores de peso, y viceversa. Aclararemos mejor esto con otros ejemplos.

Para contestar a las preguntas 2 y 3:

Escoger menú **1 fu**, *opción* **Ecuación de regresión...**

Observamos que:

Coeficiente de Correlación : ______.

Ecuación de la recta de regresión:______.

El coeficiente de correlación es un valor entre 1 y -1. Cuando más cerca se encuentre de ellos más alta será la relación entre las dos variables. Más adelante hablaremos de ello.

La ecuación de la recta es la contestación a la tercera pregunta.

Para contestar a la pregunta 4, por ejemplo: ¿Cual es el peso esperado de una persona que mida 1.70 m.?

Escoger menú **1 fu**, *opción* **Imagen...**

Escribir 1.70.

El valor esperado es de unos ______. Kg.

2-Hagamos lo mismo con las dos tablas siguientes, Calcularemos para cada una el **Coeficiente de correlación.**

a- Sean 10 alumnos de COU seleccionados al azar. Se nos da el número de aciertos en dos ejercicios de matemáticas y filosofía:

Matemática	Filosofía
80	53
48	61
62	53
53	72
43	45
72	59
60	45
41	58
53	50
57	46

Procedemos como en el caso anterior. Observamos la gráfica y el coeficiente de correlación.

- *Mediante el portapapeles dibujar la función en este lugar.*

No parece que haya una gran relación entre los dos tipos de valores.

El coeficiente de correlación, ______.

b- Hemos sacado al azar 10 monedas de una bolsa. Medimos la antigüedad y el peso en gramos. Los resultados son los siguientes:

Antigüedad (años)	Peso (Kg)
5	9.41
9	9.45
14	9.33
17	9.34
23	9.31
31	9.26
35	9.22
42	9.21
46	9.15
50	9.08

- Mediante el portapapeles dibujar la función en este lugar.

Procedemos como en el caso anterior. Observamos la gráfica y el coeficiente de correlación.

El coeficiente de correlación, ______.

En este caso sí parece que hay una gran relación entre los dos tipos de valores. Pero es distinta a más años menos peso. El coeficiente de correlación es negativo y cercano a -1

c- De todo ello llegamos a la siguiente conclusión si dos variables no tienen absolutamente ninguna relación el coeficiente de correlación tendrá un valor cercano a cero. Si hay mucha relación un valor absoluto cercano a uno. Si el coeficiente es positivo la correlación será directa, es decir más valor x mayor y, y viceversa. Si el coeficiente es negativo la correlación será inversa, mayor valor de la x implica menor valor de la y.

3- Puede que dos variables estén relacionadas, pero no necesariamente a través de una relación lineal. lo veremos con el siguiente ejemplo:

Dejamos caer una piedra en un pozo y anotamos en distintos instantes de tiempo el espacio recorrido, obtenemos la siguiente tabla:

Tiempo (seg.)	Espacio (metros)
0	0
0.5	1.25
1	5
1.5	11.25
2	20
3	45
5	125
6	180
8	320

Observamos que hay una cierta relación. De hecho el coeficiente de correlación lineal es muy bueno 0.96.

Probemos otro tipo de ajuste.

Nos dirigimos al cuadro **Regresión. Introducir Valores**.

Pulsamos el botón **Cuadrática**.

Finalmente pulsamos el botón **Aceptar**.

- Mediante el portapapeles dibujar la función en este lugar.

Observamos que una parábola ajusta perfectamente los pares de valores. El coeficiente de correlación parabólico es ______, lo que significa que los datos se ajustan perfectamente a una parábola.

FICHA 4

SOFTWARE: FW (funciones para Windows)

TEMA: Serie de Fourier

Matemática II

<u>**Tema**</u>:

Estudio de las series de Fourier.

<u>**Nivel**</u>:

Segundo año de la carrera de contador público y licenciado en administración

<u>**Conocimientos previos**</u>:

Conocer las funciones trigonométricas y, en general, el concepto de funciones periódicas. Podría ser interesante conocer el cálculo de primitivas, pero, como para calcular integrales definidas utilizaremos el mismo programa funciones, no lo creemos imprescindible.

<u>**Objetivo general**</u>:

Conocer que se puede aproximar una función periódica mediante una suma de funciones trigonométricas.

Procedimiento:

- *Ejecutar el programa:* **Funciones para Windows**.

Valores de los ejes, los iniciales.

Condiciones de trabajo:

- Gráfico minimizado.

- La opción "**Baja precisión**", del menú "**Opciones**", **No activa**. .

- La opción "**Trazar cálculos**", del mismo menú, **No activa**. Aquí el programa realiza muchos cálculos y precisamos de la máxima rapidez.

La función que estudiaremos es la de **diente de sierra**. Utilizaremos los siguiente valores:

x	F(X)
-10	-2
-6	2
-5.9999	-2
-2	2
-1.9999	-2
2	2
2.0001	-2
6	2
6.0001	-2
10	2

- Recordemos que podemos copiar los datos en el portapapeles y pegarlos en el cuadro, **FUNCIONES NUMÉRICAS - Introducir valores**.

- *Activar la opción* **Lineal**. Muy importante.

*Es una función periódica de período:*______________________.

La **serie de Fourier** o **desarrollo de Fourier** de F(X) se define por:

ao + Sumatorio[1, infinito] { an.cos(n.pi.x/L) + bn.sen(n.pi.x/L) }

donde los coeficientes de Fourier son:

an = 1/L . Integral definida[-L,L] { f(x) . cos(n.pi.x/L) }

bn = 1/L . Integral definida[-L,L] { f(x) . sen(n.pi.x/L) }

Donde **2L** es el valor del periodo y **n=0,1,2,3,...**

Lo que vamos a realizar es superponer los sucesivos desarrollos de la serie de Fourier sobre la función diente de sierra.

Para ello, debemos calcular previamente los coeficientes de Fourier, **an**, **bn**. Podemos hacerlo calculando las integrales definidas correspondientes o utilizando la opción, cálculo de **Integral definida**, que nos ofrece el programa.
A continuación, describiremos cómo hacerlo.

- *Ejecutar de nuevo el programa:* **Funciones para Windows**.

Valores de los ejes, los iniciales.

Los términos **an** valen 0 en todos los casos. La función Cos(x) es par y la función x es impar. El producto de ambas es impar. Como hemos de calcular la integral definida entre -L y L, ésta será siempre 0. Se puede comprobar calculando la integral definida de la función:
f(x)=x.cos(n.p.x/2), entre -2 y 2, siendo n cualquier número natural.

NOTA: Creemos que puede ser muy instructivo hacer esto último cada vez que se calcula un coeficiente. Observaremos cómo la aproximación va mejorando sucesivamente.

Calcular el término **b1**.

- *Escribir la función,* **f(x)=1/2x.sen(px/2)**.

- *Representarla.* Botón **Aceptar**.

Menú **1 fu.**. *Calcular la integral definida entre -2 y 2*

Repetimos el proceso para calcular los sucesivos coeficientes.

Término **b2**.

- *Escribir la función,* **f(x)=1/2x.sen(2px/2)**.

*Calcular la integral definida entre -2 y 2 =*______________________.

Término **b3**.

- *Escribir la función,* **f(x)=1/2x.sen(3px/2)**.

Calcular la integral definida entre -2 y 2 = ______________________.

Término **b4**.

- *Escribir la función,* **f(x)=1/2x.sen(4px/2)**.

*Calcular la integral definida entre -2 y 2 =*______________________.

Término **b5**.

- *Escribir la función,* **f(x)=1/2x.sen(5px/2)**.

*Calcular la integral definida entre -2 y 2 =*______________________.

El desarrollo de Fourier, para los coeficientes que hemos calculado, queda:

G(x)=____.

Volvemos a la ventana donde tenemos representada la función **diente de sierra**. Vamos al cuadro de diálogo: FUNCIONES - Entrada de datos. En el cuadro de entrada G(X), introducimos la anterior función.

- *Pulsamos el botón,* **Aceptar**. Se dibujan las dos funciones.

- *Mediante el portapapeles dibujar la función en este lugar.*

FICHA 5

SOFTWARE: MATHEMATICA

TEMA: Matrices

Mathematica

INTRODUCCIÓN

Mathematica es un sistema general diseñado por Stephen Wolfram para aplicaciones matemáticas, entre otras. Es un programa que permite desarrollar las principales las operaciones numéricas, simbólicas y gráficas, y permite manejarlas de forma unificada.

Mathematica utiliza expresiones simbólicas para proveer una expresión general de estructuras matemáticas, la generalidad de las expresiones simbólicas permite cubrir una amplia gama de aplicaciones con un número restringido de metodologías, de forma que simplifica el trabajo de diseño de las fórmulas.

Este programa ofrece la posibilidad de desarrollar gráficas tanto en dos como en tres dimensiones, así como gráficos de contorno y de densidad, empleando tanto funciones como listas de datos sobre funciones previas. En tres dimensiones, permite controlar sombreados, color, iluminación, brillo de las superficies y otras características. Varias versiones de éste sistema respaldan gráficos con animación, y pueden transferirse a una amplia variedad de programas.

Mathematica permite realizar funciones de funciones, de manera que las iteraciones obtenidas crean imágenes muy próximas a la dimensión fractal. Pueden especificarse sistemas de puntos coordenados de manera que éstos constituyan funciones de otros parámetros, construyendo con ello gráficas de tres dimensiones.

Además, el programa permite combinar segmentos de gráficas distintas, lo que en el presente trabajo permitió ligar segmentos distintos correspondientes a diferentes objetos , o bien, puede utilizarse como lenguaje de gráficas en el cual construir gráficos a partir de diferentes componentes.

Tema:

Fundamentos del álgebra lineal

Conocimientos previos:

Matrices. Algebra de matrices

Inserción curricular:

Álgebra lineal

Objetivos:

Conocer los conceptos de: Matriz. Operaciones con matrices.

Reconocer los distintos tipos de matrices y sus operaciones.

Procedimiento:

Definición de Matriz

Una matriz es un arreglo de números ordenados rectangularmente y representan entre sí un sistema lineal.

Ejecutar el programa Mathematica

La forma de definir matrices dentro de Mathematica es muy simple:

{{*a*,*b* },{*c*,*d* }} *Genera un arreglo matricial.*

Podemos en cualquier momento definir un elemento de una matriz.

<Matriz> [[*<Fila>*, *<Columna>*]] *Define el elemento posicionado en fila y columna de la matriz indicada.*

De la misma forma podemos definir una fila de la matriz.

<Matriz>> [[*<Fila>*]] *Define la fila de la matriz indicada.*

En matemática se acostumbra a colocar los elementos entre corchetes grandes. Mathematica simula este despliegue con la siguiente instrucción:

MatrixForm[*<Matriz>*]

Observemos el siguiente ejemplo:

In[4]:=A={{4,0,6}, {1,6,2},{3,9,0}}

Out[4]= {{4,0,6}, {1,6,2}, {3,9,0}}

In[5]:= A[3,2]

Out[5]= 9

In[6]:=MatrixForm[A]

Out[6]//MatrixForm = 4 0 6
1 6 2
3 9 0

Matriz Identidad

La matriz identidad es el elemento neutro en la multiplicación de matrices, es decir, que una matriz multiplicada por ésta nos dará la misma matriz; comúnmente, suele simbolizarse a esta matriz con la letra "I".
El comando para obtener dicha matriz se presenta a continuación:
IdentityMatrix[*<Dimensión>*] donde "dimensión" es la dimensión de la matriz tanto de renglones como columnas. Veamos un ejemplo:

In[7]:= IdentityMatrix[3]

Out[7]= {{1, 0, 0}, {0, 1, 0}, {0, 0, 1}}

In[8]:= MatrixForm[%]

Out[8]//MatrixForm = 1 0 0
0 1 0
0 0 1

Matriz Diagonal

Una matriz diagonal es aquella que tiene la forma interna de una diagonal, es decir, los valores diferentes de 0 se encuentran en la diagónal principal de la matriz.

Su forma de obtención es mediante el siguiente comando:

DiagonalMatrix[<*Valores*>] donde los valores son los componentes de la diagonal principal.Veamos un ejemplo de lo expuesto:

In[9]:= DiagonalMatrix[3, -4, 2]

Out[9]= {{3, 0, 0}, {0, -4, 0}, {0, 0, 2}}

In[10]:= MatrixForm[%]

Out[10]//MatrixForm=3 0 0
0 -4 0
0 0 2

Matriz Transpuesta

La matriz transpuesta se obtiene al intercambiar las filas por columnas dentro de la matriz. Así, el elemento cuya posición es [2,1] será [1,2], la [3,2] será la [2,3], etc.

Para obtener la matriz transpuesta de una matriz utilizando Mathematica se recurre al siguiente comando:

Transpose[<*Matriz*>]

Observemos este ejemplo:

In[11]:=A={{5,3,6},{2,0,7},{8,1,0}}

Out[11]= {{5, 3, 6},{2, 0, 7},{8, 1, 0}}

In[12]:=Transpose[A]

Out[12]= {{5, 2, 8}, {3, 0, 1}, {6, 7, 0}}

In[13]:= MatrixForm[%]

Out[13]//MatrixForm= 5 2 8

3 0 1

6 7 0

In[14]:= MatrixForm[Transpose[A]]

Out[14]//MatrixForm= 5 3 6

2 0 7

8 1 0

En la entrada 14 se construye un comando compuesto basado en el comando "MatrixForm" cuyo argumento es el comando "Transpose" y este a su vez, tiene como argumento la matriz A.

Matriz Inversa

Se dice que una matriz es inversa de una dada, si al inversa multiplicarse (a izquierda y derecha) por la matriz original nos da como resultado la matriz identidad. Por lo tanto:

$$A^{-1} \times A = A \times A^{-1} = I$$

Nota: para que una matriz sea inversible o regular (que admita inversa) deberá ser cuadrada.

Para obtener dicha matriz se emplea el siguiente comando:

Inverse[<*Matriz*>]

Analicemos este ejemplo:

In[15]:= W={{-4,9},{1,-2}}

Out[15]= {{-4,9},{1,-2}}

In[16]:= Inverse[W]

Out[16]= {{2,9},{1,4}}

In[17]:= MatrixForm[%]

Out[17]//MatrixForm= 2 9

1 4

Podemos observar la claridad del uso de este comando en la entrada 16. La comprobación de la definición de la matriz inversa se deja como ejercicio, pues es necesaria la multiplicación de matrices que se estudiará en la siguiente sección.

Operaciones entre matrices

Las operaciones que se pueden realizar entre matrices reciben el mismo nombre y símbolo que las de la aritmética elemental.

+	Adición matricial.
-	Sustracción matricial.
*	Multiplicación matricial (no conmutable).
/	División matricial.

Cabe indicar que la multiplicación entre matrices cumple con las siguientes características:

- Sus elementos (matrices) no se pueden conmutar.
- El número de columnas de la primera matriz debe ser igual al número de filas de la segunda matriz.

Observemos este ejemplo:

```
In[18]:= A={{-2,1,3},{4,0,-1}}

Out[18]= {{-2,1,3},{4,0,-1}}

In[19]:= B={{4,1,-2},{5,-1,3}}

Out[19]= {{4,1,-2},{5,-1,3}}

In[20]:= MatrixForm[A]

Out[20]//MatrixForm= -2  -1 –3
                     -4   0 -1

In[21]:= MatrixForm[B]

Out[21]//MatrixForm= 4  -1  -2
                     5  -1  -3

In[22]:= MatrixForm[A+B]

Out[22]//MatrixForm= 2  -2  -1
                     9  -1  -2

In[23]:= MatrixForm[A-B]

Out[23]//MatrixForm=  -6  0  5
                      -1  1  4

In[24]:= MatrixForm[A*B]
```

Out[24]//MatrixForm= -8 -1 -6

20 0 -3

Se puede apreciar que el argumento del comando MatrixForm es la operación, hecho que nos es de mucha utilidad.

Potencia de una matriz

Una operación muy común es elevar una matriz a la "n" potencia. Existe una fórmula para resolver dicho cálculo pero en Mathematica utilizaremos el comando siguiente:

MatrixPower[<*Matriz*>, <*Potencia*>]

Enfoquémonos en el siguiente ejemplo:

```
In[26]:=MatrixForm[Q={{4,3,2},{-4,5,-2},{-1,3,5}}]
Out[26]//MatrixForm= 4   3   2
                    -4   5  -2
                    -1   3   5
In[27]:= MatrixPower[%,4]
Out[27]={{-1370,621,-696},{282,-1829,-1080},
      {-1317,-45,-719}}
In[28]:= MatrixForm[%]
Out[28]//MatrixForm=  -1370    621    -696
                        282  -1829   -1080
                      -1317    -45    -719
```

Es notorio la inmensa utilidad que nos da el hecho de pasar como argumento la asignación de una matriz al comando MatrixForm como se puede apreciar en la entrada 26.

FICHA 6

SOFTWARE: MATHEMATICA

TEMA: Determinantes

Tema:

Determinantes y una aplicación (Autovalores y Autovectores)

Conocimientos previos:

Determinantes. Propiedades de los determinantes. Autovalores. Autovectores

Inserción curricular

Fundamentos del álgebra lineal

Objetivos:

Conocer los conceptos de: Determinante. Autovalores. Autovectores

Reconocer las propiedades de los Determinantes.

Comprender y saber aplicar, los Autovalores y Autovectores.

Procedimiento:

Obtención de determinantes

El determinante de una matriz es un valor asociado a la matriz misma y se define de manera inductiva. Este valor nos sirve para varias aplicaciones en álgebra lineal, entre ellas, el conocimiento del número de soluciones que una matriz (sistema de ecuaciones) pueda tener.

La forma de obtener la determinante es por medio de la siguiente instrucción:

Det[<*Matriz*>]

Veamos este ejemplo:

In[29]:= MatrixForm[A={{2,-5,3},{1,3,4},{-2,3,7}}]

Out[29]= MatrixForm= - 2 -5 3

1 3 4

-2 3 7

In[30]:= Det[A]

Out[30]= 120

Autovalores y Autovectores

Sea **A** una matriz cuadrada con elementos reales. El número λ (real o complejo) recibe el nombre de autovalor de **A** si existe algún vector diferente de cero tal que: **A.v** = λ.**v**; se dice que el vector **v** (distinto de cero0 es un autovector de **A**

La obtención de los mencionados autovalores se logra gracias al siguiente comando:

Eigenvalues[<*Matriz*>]

De la misma forma, los autovectores se resuelven con la siguiente instrucción:

Eigenvectors[<*Matriz*>]

Veamos un ejemplo:

```
In[11]:= Eigenvalues[{{a,b},{-b,2a}}]

           3a+Sqrt[9a-4(2a+b)      3a-Sqrt[9a-4(2a+b)
Out[11]= {------------------------,------------------------}
                  2                          2


In[12]:= m={{2.3,4,5},{6.7,-1.2}}

Out[12]= {{2.3,4,5},{6.7,-1.2}}

In[13]:= Eigenvalues[m]

Out[13]= {6.31303, -5.21303}

In[14]:= Eigenvectors[m]

Out[14]={{0.746335,0.66557},{-0.523116,0.873374}}

In[15]:= f={{0,1,0},{0,0,1},{0,0,0}}

Out[15]= {{0,1,0},{0,0,1},{0,0,0}}

In[16]:= Eigenvalues[f]

Out[16]= {0, 0, 0}

In[17]:= Eigenvectors[f]

Out[17]= {{1, 0, 0},{0, 0, 0},{0, 0, 0}}
```

FICHA 7

SOFTWARE: MATHEMATICA

TEMA: Vectores

Tema:

Fundamentos del álgebra lineal

Conocimientos previos:

Vectores. Álgebra de vectores. Operadores

Inserción curricular:

Álgebra lineal

Objetivos:

Conocer el concepto de vector. Operaciones con vectores.

Reconocer los distintos tipos de operaciones vectoriales.

Comprender y saber aplicar operadores vectoriales.

Procedimiento:

Análisis Vectorial

DEFINICION DE VECTOR

Un vector es una magnitud cuya determinación exige el conocimiento de un módulo, una dirección y un sentido.

Ahora bien, todo vector posee vectores componentes que están en la dirección de los vectores unitarios. Estos vectores unitarios tienen como magnitud a la unidad y son aquellos que son paralelos a un positivo.

Es así entonces que la magnitud en *x* de un vector tiene la dirección del vector unitario i, *y* la tiene en j y *z* la tiene en k.

En Mathematica, los vectores se escriben de la siguiente forma:

{*<Magnitud en i>*, *<Magnitud en j >*, *<Magnitud en k>* }

Analicemos este ejemplo:

```
In[1]:= A={5,2,3}
Out[1]:= {5,2,3}
```

NOTA IMPORTANTE: Para poder correr este paquete que está incluido en Mathematica es necesario llamarlo mediante la siguiente expresión:

```
<<Calculus'VectorAnalysis'
```

Una vez ingresado podremos ejecutar cualquier operación relacionada con este tema.

Producto Vectorial

Dados los vectores A y B, su producto vectorial o externo es otro vector C=AxB. El módulo de AxB es el producto de módulos por el seno del ángulo que forman. La dirección de C es perpendicular al plano que forman A y B y su sentido es el dado por la regla de la mano derecha.

Resolvamos este problema empleando Mathematica.

CrossProduct[*<Vector 1>*, *<Vector 2>*]

Operador Nabla

El operador nabla se utiliza para denotar la derivación parcial de cada componente de un vector. Se simboliza con una delta girada 180 grados.

Gradiente

El gradiente de un vector es la multiplicación directa del operador nabla por el vector. El producto de esta multiplicación siempre será un tensor de orden 2. (Un tensor de orden 2 es aquel ente matemático que requiere de dos cantidades para especificarse)

La manera de ejecutarse esta operación tan larga en Mathematica es como se expone a continuación:

Grad[<*Vector*>]

Divergente

El divergente de un vector es el producto punto del operador nabla con el vector. El resultado de esta operación es un escalar por definición

Desarrollar este algoritmo es bastante fácil.

Div[<*Vector*>]

Rotacional

El rotacional de un vector viene dado por el producto cruz del operador nabla con el vector.

Como es costumbre en Mathematica se resuelve toda operación mediante un operador y en este caso el que se utiliza es el siguiente:

Curl[<*Vector*>]

Laplaciano

El laplaciano de un vector se define como la doble aplicación del operador nabla al vector dado.

El comando que se utiliza para este problema es el siguiente:

Laplacian[<*Función*>]

FICHA 8

SOFTWARE: MATHEMATICA

TEMA: Números Complejos

Tema:

Números Complejos

Conocimientos previos:

Números Complejos.

Inserción curricular: Fundamentos del álgebra

Objetivos: Conocer los conceptos de: Números Complejos

Reconocer las operaciones de los números complejos.

Procedimiento:

Manejo de números complejos

Existe una clasificación dentro de los números llamada números irreales o complejos y trata de aquellos números que gráficamente no existen, o sea, que no son representables pero que existen imaginariamente. Se ha escogido como símbolo de este tipo de números a la letra "i" y su valor es igual a la raíz cuadrada de menos uno. Mathematica emplea la letra "I" para reconocer dichos números.
Una expresión compleja se define como aquella que tiene un término complejo:

$$x + i\, y$$

Mathematica dispone de los siguientes comandos para poder operar con este tipo de expresiones. En realidad, una expresión compleja representa un número del mismo tipo

Re[*<Expresión compleja>*]*Extrae la parte real de la expresión.*

Im[*<Expresión compleja>*]*Extrae la parte imaginaria de la expresión.*

Conjugate[*<Expresión compleja>*]*Obtiene el complejo conjugado de la expresión.*

Abs[*<Expresión compleja>*]*Define la magnitud de la expresión.*

Arg[*<Expresión compleja>*]*Obtiene el ángulo o argumento de la forma polar de la expresión.*

Veamos el siguiente ejemplo donde se aplican los comandos expuestos:

```
In[7]:= z=Sqrt[3]-I

Out[7]:= -I + Sqrt[3]

In[8]:= Re[z]

Out[8]:= Sqrt[3]

In[8]:= Im[z]

Out[8]:= -I

In[8]:= Conjugate[z]
```

```
Out[8]:= 2I + Sqrt[3]

In[9]:= Abs[z]

Out[9]:= 2

In[10]:= Arg[z]

          -Pi
Out[10]:= ---
           6

In[11]:= N[% (1/Degree)]

Out[11]:= -30
```

FICHA 9

SOFTWARE: MATHEMATICA

TEMA: Trigonometría

Tema:

Trigonometría

Conocimientos previos: Inserción curricular: Temas de álgebra

Objetivos: Conocer los conceptos de: Funciones trigonométricas. Funciones hiperbólicas.

Reconocer las propiedades de las relaciones trigonométricas

Comprender y saber aplicar, las funciones que dependen de un ángulo.

Procedimiento:

Trigonometría

INTRODUCCION

La trigonometría es un tema importante dentro del área de la matemática que se aplica a las mediciones de las partes o elementos de un triángulo.

La trigonometría se basa en algunas relaciones, llamadas *funciones trigonométricas* que describen el comportamiento de los ángulos de un triángulo en función de sus lados.

Una de las primeras aplicaciones de la trigonometría se hicieron en la navegación y la ingeniería y después, se extendió a otros campos como por ejemplo, la astronomía.

Estas funciones también desempeñan un papel importante en toda clase de fenómenos vibratorios (sonido, luz, electricidad, etc.).

Conversión de unidades de medición angular

Un ángulo es la abertura que forman dos rectas al intersecarse.

Existen dos unidades muy comunes para medir dicho ángulo:

Grados sexagesimales

Radianes

Es posible que deseemos convertir un ángulo medido en grados a radianes. Esto se logra en Mathematica con la instrucción siguiente:

<*Grados*> Degree

En realidad lo que se está realizando matemáticamente es obtener la razón de proporción entre la constante Pi y la mitad del valor angular de un círculo. Veamos:

Un grado $\dfrac{\pi}{180\,\text{grados}}$ Radianes

Observemos este caso de conversión:

```
In[1]:=30Degree
Out[1]:=30Degree
In[2]:=N[30Degree]
Out[2]:=0.523599
```

Como vemos, es necesario teclear el filtro de exactitud "N" para obtener el valor en cuestión.

Inversamente podemos convertir radianes a grados de la siguiente forma:

<*Radianes*> (1/Degree)

In[3]:=N[2Pi(1/Degree)]

Out[3]:=360

In[4]:=N[pi/4(1/Degree)]

Out[4]:= 45

Funciones trigonométricas

Las funciones trigonométricas son relaciones que involucran a los catetos y a la hipotenusa de un triángulo. Existen seis funciones cada una con su respectiva escritura en Mathematica:

Sin	Función seno.
Cos	Función coseno.
Tan	Función tangente.
Cot	Función cotangente.
Sec	Función secante.
Csc	Función cosecante

De igual forma, las funciones trigonométricas inversas se definen fácilmente.

ArcSin	Función arcoseno.
ArcCos	Función arcocoseno.
ArcTan	Función arcotangente.
ArcCot	Función arcocotangente.
ArcSec	Función arcosecante.
ArcCsc	Función arcocosecante.

Recuérdese que los argumentos se escriben entre corchetes seguidos de los comandos, en este caso, las funciones. Veamos este ejemplo de como Mathematica reconoce expresiones trigonométricas:

```
In[5]:= Sin[x]^2
            2
Out[5]:= Sin[x]
In[6]:= N[Sin[60],5]
Out[6]:= 0.86602
```

Una confusión podría surgir en la entrada 1. Se podría pensar que es el argumento el que se eleva al cuadrado, sin embargo, es la función la que se eleva a la potencia mencionada. Si se desea lo primero la escritura sería así:

In[6]:= Sin[x^2]

Funciones Hiperbólicas

Las funciones hiperbólicas son las siguientes:

Sinh	Función seno hiperbólico.
Cosh	Función coseno hiperbólico.
Tanh	Función tangente hiperbólica.
Coth	Función cotangente hiperbólica.
Sech	Función secante hiperbólica.
Csch	Función cosecante hiperbólica.

Comandos trigonométricos

Así como las funciones relacionan elementos (catetos e hipotenusa), las identidades relacionan funciones manteniendo cierta interacción con los componentes de un triángulo. Es decir, es posible escribir una función en términos de cualquier otra función.

El comando que se emplea es el siguiente:

TrigReduce[<*Expresión*>]

Una lista de comandos útiles al tema se presenta a continuación:

Expand[<*Expr*>, Trig->True] *Escribe las potencias de los argumentos como coeficientes de los mismos.*

Factor[<*Expr*>, Trig->True] *Factoriza las expresiones trigonométricas.*

TrigToComplex[<*Expr*>] *Escribe una expresión trigonométrica en términos complejos utilizando las identidades de Euler.*

ComplexExpand[<*Expr*>] *Escribe expresiones complejas en términos trigonométricos.*

TrigFactor[<*Expr*>] *Descompone la expresión en factores.*

Observemos algunos ejemplos:

```
In[2]:=Expand[Sin[x]^2+Sin[2x]^2,Trig->True]
            Cos[2 x]     Cos[4 x]
Out[2]= 1 - -------- - ---------
               2           2

In[3]:= Sin[ax]/2+Sin[ax-2bx]/4+Sin[ax+2bx]/4
                        2

Out[3]= Cos[b x]  Sin[a x]

In[4]:= ComplexExpand[Tan[x+Iy]]

              Sin[2 x]                I Sinh[2 y]
Out[4]=   -----------------    +   -----------------
          Cos[2 x]+Cosh[2 y]       Cos[2 x]+Cosh[2 y]
```

FICHA 10

SOFTWARE: MATLAB

TEMA: Comandos Básicos

MATLAB

INTRODUCCIÓN

MALTLAB en un lenguaje de programación, desarrollado para resolver problemas de Análisis Numérico. Es un programa para computación numérica que permite visualizar datos. Es ampliamente usado en diversas áreas de tecnología, ya que posee además una extraordinaria capacidad para resolver problemas desde el punto de vista de la matemática aplicada, economía, física, química, ingeniería y otras aplicaciones. Está realizado a través de un sotfware de matrices para el análisis de sistemas de ecuaciones. Permite resolver problemas de análisis numéricos sin necesidad de escribir un gran programa.

MATLAB emplea matrices porque con ellas se puede describir infinidad de cosas de una forma altamente flexible y matemáticamente eficiente. Una matriz puede describir una relación lineal entre los componentes de un modelo matemático.

En este último sentido, una matriz puede describir el comportamiento de un sistema extremadamente complejo. Por ejemplo puede representar un procesamiento digital de imágenes o un filtro digital de procesamiento de señales.

COMANDOS BÁSICOS:

Una lista de alfabética de los comandos de MATLAB. Puede usarse el comando "help <comando>" para obtener más información sobre como se usa los comandos de MATLAB.

acker: calcula la matriz K la coloca en el punto extremo de A-BK

axis: determina la escala para el flujo del plot

bode: dibuja el bode plot

cloop: cierra el loop y transfiere la función

conv: circunvolución (conveniente para multiplicar polígonos)

ctrb: la controlabilidad de la matriz

cet: encontrar una determinada matriz

cig: calcula una estimación de la matriz

eps: tolerancia (esiplon de la máquina)

for: para, continuo loop (repetitivo)

format: número de formato (significa dígitos, exponentes)

grid: dibuja un conjunto de líneas cuadriculadas sobre el plot

help: ayuda

inv: encuentra el inverso de la matriz

length: longitud del vector

loglog: plot usando la escala log-log

lsim: simula líneas del sistema

norm: regla para un vector

nyquist: dibuja un Nyquist en el plot

obsv: para observar la matriz

ones: retorna un vector o una matriz una vez

place: calcula la matriz K la coloca en el punto extremo de A-BK

plot: dibuja un plot

print: imprime el plot (para una impresora o un archivo postscript)

rlocfind: encuentra el valor de K y selecciona un punto y un punto extremo

rlocus: dibuja el root locus

roots: encuentra el roots de un polinomio

set: set(gca,'Xgca,'Xtick',xticks,'Ytick',yticks) para el control de numeros y separación interlinear de etiquetar al instante sobre un axes

size: obtiene las dimensiones de un vector o una matriz

sqrt: raíz cuadrada

ss2tf (state-space): para transferir una representación de función

ss2zp (state-space): para representarlo en Pole-zero

step Plot para step (repuesta)

subplot: divide la ventana del plot en segmentos

sum: suma todos los elementos de un vector

text: añade una pieza de texto en el plot

tf2ss: traslada la función a una representación state-space tf2zp: traslada la función a una representación pole-zero

title: añade titulo al plot

xlabel: añade una etiqueta al eje de simetría horizontal en el plot

ylabel: añade una etiqueta al eje de simetría vertical en el plot

zeros: retorna una matriz o un vector de ceros

zp2ss (pole-zero): para representarlo state-space

zp2tf (pole-zero to): transferir una representación de una función

Uso básico de MATLAB

Normalmente se requiere de modelos computacionales con el fin de resolver problemas de matemática aplicada. Muchas veces puede ser útil hacer un programa que utilice matrices, complejos, y otras estructuras matemáticas, pero fácil de escribir y revisar. MATLAB es ideal para esto.

Esta presentación está organizada de la siguiente forma:

- Generalidades.
- Comandos de programación.
- Comandos matemáticos.
- Programas de ejemplo variados.

Cada uno de los vínculos de estas secciones, contiene una explicación breve y ejemplos pequeños de cada comando. La sección de ejemplos, contiene algunos programas completos, donde se utilizan los comandos tratados.

NOTA:

En todos los programas de ejemplo se utiliza el comando de MATLAB: **%** el cual se utiliza para añadir un comentario en el programa. Estos comentarios son importantes para que otros puedan entender el contenido con mayor facilidad.

Generalidades

Esta es una breve introducción al manejo de variables (escrita para las personas que nunca han usado MATLAB), expresiones y archivos con extensión .m (programas ejecutables por MATLAB), con respecto a su creación y uso.

Comandos básicos de programación

Para la estructura de programación en MATLAB se requiere conocer por lo menos los siguientes comandos:

- **END**

  Determina hasta cual orden llega el efecto de if, for, y while. (Para ejemplos de su uso ver if, while y for)

- **IF**

  Verifica si se cumple cierta condición, y de acuerdo a si se cumple o no realiza la acción que se desee.

- **WHILE**

  Realiza una parte del programa mientras se cumpla alguna condición.

- **FOR**

  Muy parecido al While, pero utiliza un contador, es útil si se quiere repetir una parte del programa un número. determinado de veces.

- **CLEAR**

  Borra todas las variables de la memoria. Es recomendable usarlo al principio de todos los programas. (simplemente escriba clear; aL comienzo del programa)

- **PLOT**

  Sirve para obtener resultados gráficos en 2D.

- **DISP**

  Sirve para escribir texto de salida o vectores. de resultados.

- **INPUT**

  Se utiliza para que el programa pida valores de variables mientras se ejecuta.

Descripción detallada de algunos de ellos

Comando IF

Verifica si se cumple cierta condición, y de acuerdo a si se cumple o no realiza la acción que se desee.

Sintaxis:

La sintaxis de la orden es:

if (condición), (ordenes 1) [else, (ordenes 2)] end;

Donde las ordenes entre [ ] son opcionales.

(ordenes 1) son las ordenes que se realizarán si (condición) se cumple.

(ordenes 2) son las ordenes que se realizarán si (condición) NO se cumple.

(condición) Puede ser:

a == b (verifica si a es igual a b)
a < b
a > b
a <= b (verifica si a es menor o igual que b)
a >= b
a ~= b (verifica que a y b sean diferentes)

Ejemplo simple de uso:

El siguiente ejemplo ilustra el uso de if:

```
%Ejemplo de uso de if.
n=0;
if n=0,
n   % al escribir una expresión sin punto y coma final, MATLAB escribe su resultado en pantalla.
else,
n=1
end;
n=2;
if n=0,
n
else,
n=1
end;
```

La salida que se obtiene con el programa anterior es la siguiente:

```
n=0
n=1
```

Donde el 0 proviene de entrar al primer if, y el uno, de entrar al else del segundo if.

Comando WHILE

Realiza una parte del programa mientras se cumpla alguna condición.

Sintaxis:

La sintaxis de la orden es:

while (condición), (ordenes) end;

(ordenes) son las ordenes que se realizarán mientras (condición) se cumpla.

(condición) Puede ser:

a == b (verifica si a es igual a b)

a < b

a > b

a <= b (verifica si a es menor o igual que b)

a >= b

a ~= b (verifica que a y b sean diferentes)

Ejemplo simple de uso:

El siguiente ejemplo ilustra el uso de while:

```
%Ejemplo de uso de while.
n=0;
while n<=5,
n    %Al escribir el nombre de la variable (sin punto y coma) MATLAB imprime su valor.
n=n+1; %El punto y coma evita que MATLAB imprima el nuevo valor de n.
end;
```

La salida que se obtiene al correr el programa anterior es:

n= 0

n=1

n=2

n=3

n=4

n=5

Comando FOR

Muy parecido al While, pero utiliza un contador, es útil si se quiere repetir una parte del programa un número determinado de veces.

Sintaxis:

La sintaxis de la orden es:

for (contador), (ordenes) end;

(ordenes) son las ordenes que se realizarán (contador) llega a su valor final.

(contador) Es de la forma:

variable = a [,b] : c

Donde:

- variable es el contador en sí.
- a es el valor inicial del contador (variable).
- b es el segundo valor del contador (opcional, si se omite, b=a+1), su función es determinar el incremento del contador.
- c es el valor final del contador (variable).

Ejemplo simple de uso:

El siguiente ejemplo ilustra el uso de for:

```
%Ejemplo de uso de for.
for i=0,0.5:2.5,
i  %al escribir el nombre de una variable (sin punto y coma) MATLAB muestra su
valor.
end;
```

La salida del programa anterior es la siguiente:

i=0

i=0.5

i=1

i=1.5

i=2

i=2.5

Comando PLOT

Sirve para obtener resultados gráficos en 2D.

Sintaxis:

La sintaxis de la orden es:

plot(x, y);

x es el vector que contiene los valores de x.
y es el vector que contiene los valores de y, tal que el valor de y en la posición uno del vector corresponde al primer valor del vector x. La gráfica se realiza uniendo una serie de rectas entre los puntos incluidos en los vectores X y Y. Si las curvas quedan muy mal hechas (se notan las rectas) puede ser necesario disminuir el paso de los vectores y aumentar el número de puntos.

Para claridad, puede ser necesario leer la parte correspondiente a VECTORES y a la orden FOR.

Ejemplo simple de uso:

El siguiente ejemplo ilustra el uso de plot:

```
%Ejemplo de uso de plot.
for i=1:101,
x(i)=(i-1)/100;
y(i)=x(i)+1;  % Organiza en vectores la función y=x+1
end;
plot(x, y);
pause;
%pausa el computador hasta que se presione una tecla esta orden es necesaria
%cuando se hace más de una gráfica, para poder ver cada una por separado. Ya
que %MATLAB las dibuja en la misma ventana siempre (a menos que se use el
comando %FIGURE).
```

Al correr el programa se obtiene la gráfica de la recta y=x+1 (para 0<=x<=1). La gráfica aparecerá en una ventana aparte llamada Figure 1.

Comando DISP

Sirve para escribir texto de salida o vectores (y matrices) sin mostrar su nombre.

Sintaxis:

La sintaxis de la orden es:

disp(X);

X Puede ser:

- Un vector.
- Una matriz.
- Una cadena de texto.

Ejemplo simple de uso:

El siguiente ejemplo ilustra el uso de disp:

```
%Ejemplo de uso de disp.
a=[1, 2, 3, 4]; % Un vector
disp(a);
a=[1, 2; 3, 4]; % Una matriz
disp(a);
a='Texto para escribir'; % Cadena de texto
disp(a);
disp('También se puede usar así.');
```

La salida del programa anterior será:

```
1       2       3       4

1       2
3       4
```

Comando INPUT

Se utiliza para que el programa pida valores de variables mientras se ejecuta.

Sintaxis:

La sintaxis de la orden es:

variable = input(texto);

variable es un nombre válido de variable, en la que se quiere almacenar el valor que se pregunta.

El texto puede ser:

- Una variable
- Una cadena.

Ejemplo simple de uso:

El siguiente ejemplo ilustra el uso de input:

```
%Ejemplo de uso de input.
a=0;  % hace válido el nombre de variable a.
a=input('Teclee el valor de a: ');
tex='Cual es el nuevo valor de a? ';
a   % Al escribir el nombre de una variable (sin punto y coma al final)
% MATLAB muestra su valor.
a=input(tex);
a
```

La salida de este programa será:

Teclee el valor de a: (espera)

a=xxx

Cual es el nuevo valor de a? (espera)

a=yyy

Donde xxx y yyy son valore introducidos por el usuario en el momento de correr el programa.

FICHA 11

SOFTWARE: MATLAB

TEMA: Matrices y Vectores

VECTORES Y MATRICES

Asignación de valores y subíndices:

Los vectores y matrices en MATLAB se trabajan igual en cuanto a asignación, por eso se explican juntos. Pero las operaciones posibles, si son diferentes, y están separadas bajo los encabezados correspondientes.

Asignación:

La asignación de variables en MATLAB es sencilla, y los vectores y matrices no son la excepción. Cuando se desea dar el valor a toda una matriz se puede realizar directamente de la siguiente forma:

```
A=[1 2 3 4; 5 6 7 8;9 0 1 2]; ó

A=[1, 2, 3, 4;5, 6, 7, 8;9, 0, 1, 2];
```

donde la matriz escrita arriba es:

1	2	3	4
5	6	7	8
9	0	1	2

Las filas se separan por punto y coma y las columnas por espacios o comas. De lo anterior se ve fácilmente que un vector fila se asigna así:

```
v=[1 2 3];     ó     v=[1, 2, 3];
```

Y un vector columna se asigna así:

```
v=[1; 2; 3];
```

Manejo de subíndices:

Otra forma de asignar valores a una matriz (o un vector) es por medio de los subíndices. El menor subíndice utilizado por MATLAB es 1. Y va añadiendo valores a medida que se requieran. Los subíndices se escriben entre paréntesis. Por ejemplo:

A(2, 3)=1; Asigna al elemento en la fila 2, columna 3 el valor de 1.

Si se desea cambiar todo el valor de una fila o una columna, puede hacerse con el operador ":" así:

A(1,:)=[4 5 6];

Asigna a la fila 1 el vector [4, 5, 6] (cambia la fila 1 por 4, 5, 6). Así si A era una matriz de 3x3 de ceros, ahora queda:

4	5	6
0	0	0
0	0	0

Igualmente a veces se requiere trabajar con vectores que son una columna o una fila de una matriz. Esto se realiza fácilmente guardando este "vector" en un vector, así:

v=A(:,1);

Asigna al vector v la primera columna (completa) de la matriz A.

Operaciones matemáticas simples con matrices y vectores:

Esto es algo en lo que MATLAB hace las cosas verdaderamente simples, si se tienen dos matrices (o vector y matriz, o dos vectores), y se quieren: sumar, multiplicar ó restar sólo es necesario anotar esta operación normalmente (como se haría con números). Por ejemplo:

Si se quieren multiplicar dos matrices A y B y almacenar el resultado en C:

C=A*B; (Si se hace entre dos vectores (uno fila y el otro columna) el resultado es el producto punto entre los dos)

Si se quieren sumar ó restar y almacenar el resultado en C:

C=A+B; ó C=A-B; (Sin importar que sean matrices o vectores.)

Comandos matemáticos para matrices:

Los comandos matemáticos más empleados con matrices son:

<u>**NORM**</u>

Calcula la norma de un vector o matriz.

<u>**MIN**</u>

Retorna el (los) menor (es) componente (s) de un vector o matriz.

<u>**MAX**</u>

Retorna el (los) mayor (es) componente (s) de un vector o matriz.

<u>**SIZE**</u>

Devuelve las dimensiones de la matriz.

<u>**EIG**</u>

Calcula los valores y vectores propios (autovalores y autovectores) de la matriz.

<u>**INV**</u>

Invierte la matriz. (si es posible)

<u>**DET**</u>

Calcula el determinante de la matriz.

Comandos matemáticos para vectores:

Los comandos matemáticos más empleados con vectores son:

<u>**NORM**</u>

Calcula la norma de un vector o matriz.

<u>**MIN**</u>

Retorna el (los) menor (es) componente (s) de un vector o matriz.

<u>**MAX**</u>

Retorna el (los) mayor (es) componente (s) de un vector o matriz.

<u>**CROSS**</u>

Calcula el producto cruz entre vectores.

<u>**LENGTH**</u>

Determina el número de componentes de un vector.

Descripción detallada de algunos de ellos

Comando NORM

Calcula la norma de un vector o matriz.

Sintaxis:

La sintaxis de la orden es:

Norma = norm(Matriz [, Tipo]);

Los signos [ ] son para decir que Tipo es opcional.

Matriz es la matriz o vector al que se desea calcular la norma.

Tipo es el tipo de norma que se desea calcular. Tipo puede ser una de las siguientes:
Si se omite: calcula la norma 2 en un vector es la magnitud del vector 2: calcula la norma 2

inf: calcula la norma infinito
en un vector es el máximo valor absoluto
en una matriz es la suma más grande de las filas.

Hay más, pero las anteriores son las más utilizadas.

En Norma se almacena el valor de la norma calculada.

Ejemplo simple de uso:

El siguiente ejemplo ilustra el uso de norm: (ver orden de programación DISP)

```
%Ejemplo de uso de norm.
A=[1 2; 3 4]
v=[1 2 3 4]
disp('Para la matriz:');
n2=norm(A)
ni=norm(A, inf)
disp('Para el vector:');
n2=norm(v)
ni=norm(v, inf)
% Al escribir una expresión sin punto y coma al final
% MATLAB muestra su valor en pantalla.
```

Al correr el programa se obtienen como salida los siguientes resultados:

```
A =    1    2
       3    4



v =    1    2    3    4

Para la matriz:

n2 =  5.4650


ni =  7

Para el vector:

n2 =  5.4772
ni =    4
```

Comando MIN

Retorna el (los) menor (es) componente (s) de un vector o matriz. Para el caso de los vectores: retorna el menor valor contenido en sus componentes. En el caso de una matriz MIN retorna un vector (fila) que contiene el mínimo elemento que se encontró en cada una de las columnas (la primera componente del vector tiene el menor elemento en la primera columna de la matriz, y así sucesivamente).

Sintaxis:

La sintaxis de la orden es:

Mínimo = min(matriz);

Matriz es la matriz o vector al que se desea encontrar la (s) mínima (s) componente (s).
En Mínimo se retorna (n) el (los) mínimo (s) valor (es) encontrado (s) en la matriz o vector.

Ejemplo simple de uso:

El siguiente ejemplo ilustra el uso de min:

```
%Ejemplo de uso de min.
A=[1 2; 3 4]
v=[1 2 3 4]
M=min(A)
m=min(v)  % MATLAB diferencia entre mayúsculas y minúsculas.
% Al escribir una expresión sin punto y coma al final
% MATLAB muestra su valor en pantalla.
```

Al correr el programa anterior se obtiene como salida lo siguiente:

```
A =   1   2
      3   4

v =   1   2   3   4

M =   1   2
m =   1
```

Comando MAX

Retorna el (los) mayor (es) componente (s) de un vector o matriz. Para el caso de los vectores: retorna el mayor valor contenido en sus componentes. En el caso de una matriz MAX retorna un vector (fila) que contiene el máximo elemento que se encontró en cada una de las columnas (la primera componente del vector tiene el mayor elemento en la primera columna de la matriz, y así sucesivamente).

Sintaxis:

La sintaxis de la orden es:

Máximo = max(Matriz);

Matriz es la matriz o vector al que se desea encontrar la (s) máxima (s) componente (s).
En Máximo se retorna (n) el (los) máximo (s) valor (es) encontrado (s) en la matriz o vector.

Ejemplo simple de uso:

El siguiente ejemplo ilustra el uso de max:

```
%Ejemplo de uso de max.
A=[1 2; 3 4]
v=[1 2 3 4]
M=max(A)
m=max(v)  % MATLAB diferencia entre mayúsculas y minúsculas.
% Al escribir una expresión sin punto y coma al final
% MATLAB muestra su valor en pantalla.
```

Al correr el programa anterior se obtiene como salida lo siguiente:

```
A =    1    2
       3    4

v =    1    2    3    4

M =    3    4
m =    4
```

Comando SIZE

Devuelve el tamaño de la matriz (dimensiones).

Sintaxis:

La sintaxis de la orden es:

[Filas, Columnas] = size(Matriz); (los símbolos [ ] se escriben.)

ó también: **Tamaño = size(Matriz);**

Matriz es la matriz a la que se le desea determinar el tamaño (dimensiones).

En Filas se almacena el número de filas.
En Columnas se almacena el número de columnas.

Tamaño es un vector (fila) en cuyas componentes se almacenan el número de filas y de columnas, siempre en ese orden.

Ejemplo simple de uso:

El siguiente ejemplo ilustra el uso de size:

```
%Ejemplo de uso de size.
A=[1 2 3; 4 5  6]
y=size(A)
[f, c]=size(A);
f  % Al escribir una expresión sin punto y coma final MATLAB
c  % muestra el valor por pantalla
```

Al correr el programa se obtiene la siguiente salida:

```
A =   1    2    3
      4    5    6

y =    2    3

f =    2

c =    3
```

Comando EIG

Calcula los valores y vectores propios (autovalores y autovectores) de la matriz.

Sintaxis:

La sintaxis de la orden es:

[Vectores, Diagonal] = eig(Matriz); (los símbolos [ ] se escriben.)

ó también: **Valores = eig(Matriz);**

Matriz es la matriz (cuadrada) a la que se le desea calcular los autovalores o autovectores.

Diagonal es una matriz diagonal que contiene los autovalores de Matriz.

Vectores es una matriz en la que se devuelven los autovectores (unitarios) donde cada columna de la matriz es un autovector de matriz; tal que el primer vector corresponde al primer autovalor y así sucesivamente.

Valores es un vector columna que contiene los autovalores de Matriz.

Ejemplo simple de uso:

El siguiente ejemplo ilustra el uso de eig:

```
%Ejemplo de uso de eig.
A=[1 2; 3 4]
y=eig(A)
[V, D]=eig(A);
V  %Al escribir una expresión sin punto y coma final MATLAB muestra el valor por pantalla
D
```

Al correr el programa se obtiene la siguiente salida:

```
A =   1   2
      3   4

y =  -0.3723
      5.3723

V =  -0.8246  -0.4160
      0.5658  -0.9094

D =  -0.3723        0
          0    5.3723
```

Comando INV

Sirve para invertir una matriz.

Sintaxis:

La sintaxis de la orden es:

matriz1 = inv(matriz2);

matriz2 es la matriz que se desea invertir

En matriz1 se almacena la matriz inversa de matriz 2.

Ejemplo simple de uso:

El siguiente ejemplo ilustra el uso de inv:

```
%Ejemplo de uso de inv.
A=[1 2; 3 4]
I=inv(A);
I  % Al escribir una expresión sin punto y coma al final
% MATLAB muestra su valor en pantalla.
```

Al correr el programa se obtiene como salida la matriz que se desea invertir (A), y su inversa (I). La salida se ve así:

```
A = 1    2
    3    4

I= -2.0000   1.0000
    1.5000   -0.5000
```

Comando DET

Calcula el determinante de una matriz.

Sintaxis:

La sintaxis de la orden es:

Valor = det(Matriz);

Matriz es la matriz (cuadrada) a la que se le desea calcular el determinante.

Valor es donde se almacena el valor del determinante.

Ejemplo simple de uso:

El siguiente ejemplo ilustra el uso de det:

```
%Ejemplo de uso de det.
A=[1 2; 3 4]
d=det(A) % Al escribir una expresión sin punto y coma final MATLAB
% muestra en pantalla su valor.
```

Al correr el programa se obtiene la siguiente salida:

```
A =    1    2
       3    4

d =    -2
```

Comando CROSS

Calcula el producto cruz entre dos vectores.

Sintaxis:

La sintaxis de la orden es:

Vector1 = cross(Vector2, Vector 3);

Vector2 y Vector3 son los vectores a los que se les quiere aplicar el producto cruz. Tanto Vector2 como Vector3 deben ser vectores tridimensionales.

Vector1 es el vector (tridimensional) resultante del producto cruz de Vector2 y Vector3.

Ejemplo simple de uso:

El siguiente ejemplo ilustra el uso de cross:

```
%Ejemplo de uso de cross.
x=[1 0 0]
y=[0 1 0]
z=cross(x, y) % Al escribir una expresión sin punto y coma final MATLAB
% muestra en pantalla su valor.
```

Al correr el programa se obtiene la siguiente salida:

```
x =    1    0    0


y =    0    1    0


z =    0    0    1
```

Comando LENGTH

Determina el número de componentes de un vector.

Sintaxis:

La sintaxis de la orden es:

Longitud = length(Vector);

Vector es el vector que se quiere medir (número de componentes).

Longitud es el número de componentes de Vector.

Ejemplo simple de uso:

El siguiente ejemplo ilustra el uso de lenght:

```
%Ejemplo de uso de length.
x=[1 2 3 4 5 6 7]
l=length(x) % Al escribir una expresión sin punto y coma final MATLAB
% muestra en pantalla su valor.
```

Al correr el programa se obtiene la siguiente salida:

```
x =     1     2     3     4     5     6     7

l=      7
```

FICHA 12

SOFTWARE: MATLAB

TEMA: Números Complejos

NÚMEROS REALES Y COMPLEJOS

Asignación de valores a variables:

Los números complejos se trabajan igual que los reales en lo que se refiere a asignación, a operaciones matemáticas y a comandos. A continuación, se dan unos pocos ejemplos para mostrar como se realiza la asignación:

a=5.2347;

b=3;

c=1+2j; ó también:

d=1.5476+2.8*i; (el uso de j ó i es indiferente, desde que se tenga en cuenta la

Nota importante sobre el uso de las variables i y j (abajo)

d=5.2347;

e=3;

Operaciones matemáticas simples:

Las operaciones simples son las siguientes:

- Suma (operador +)
- Resta (operador -)
- Multiplicación (operador *)
- División (operador /)
- Potenciación (operador ^)

A continuación hay algunos ejemplos para complejos:

a=1+2i;

b=2+j;

c=a+b dá como resultado:

c =   3.0000 + 3.0000i

d=a^b Dá como resultado:

d = -1.6401 + 0.2021i

Nota importante sobre el uso de las variables i y j:

Puede usarse indistintamente las dos variables incorporadas (i ó j) y MATLAB no pone problema si se usan las dos al tiempo. Pero si se asignan las variables i y/o j en algún lugar del programa, esta variable perderá su valor como raíz de -1. Para clarificar esto es útil un ejemplo:

```
%Observación para cuando se trabaja con complejos.

i=8

j=9

c=2+3*j
```

La salida del programa anterior es:

```
i =   8
j =   9
c =   29
```

Como se puede ver si se intentaba representar un complejo con la variable c, no se logró debido a que se cambiaron las variables i y j. Por lo tanto se recomienda que si se va a trabajar con complejos en un programa: deje libres las variables i y j (no las utilice en contadores ó en otros propósitos, que no sean representar raíz de -1.

Comandos matemáticos para números (complejos y reales):

Los comandos matemáticos más empleados con números son:

- **ABS**

  Calcula la norma de un complejo o el valor absoluto de un real.

- **SQRT**

  Calcula la raíz cuadrada de un complejo o de un real.

- **ANGLE**

  Calcula el ángulo de fase (en radianes de 0 a 2*pi) de los elementos complejos de una matriz. (Se puede usar para calcular el ángulo de fase de un solo complejo)

Descripción detallada de ellos

Comando ABS

Calcula la norma de un complejo, o el valor absoluto de un real.

Sintaxis:

La sintaxis de la orden es:

Valor = abs(Número);

Valor es la norma del complejo si (Número es complejo) o el valor absoluto de Número (si es real).

Número puede ser un real o un complejo:

Si es Real: calcula el valor absoluto.

Si es Complejo: calcula la norma del complejo.

Ejemplo simple de uso:

El siguiente ejemplo ilustra el uso de abs:

```
%Ejemplo de uso de abs.

R=-1.2341

C=1.5+3j

disp('Para un real:');

v=abs(R)

disp('Para un complejo:');

v=abs(C)

% Al escribir una expresión sin punto y coma al final
% MATLAB muestra su valor en pantalla.
```

Al correr el programa se obtienen como salida los siguientes resultados:

```
R =    1.2341
C =   1.5000 + 3.0000i

Para un real:

v =    1.2341

Para un complejo:

v =    3.3541
```

Comando SQRT

Calcula la raíz cuadrada de un complejo o de un real.

Sintaxis:

La sintaxis de la orden es:

Valor = sqrt(Número);

En Valor se almacena la raíz cuadrada del número.

Número puede ser un real o un complejo (si es real negativo, el resultado es un complejo)

Ejemplo simple de uso:

El siguiente ejemplo ilustra el uso de sqrt:

```
%Ejemplo de uso de sqrt.

R=-1.2341

raiz=sqrt(R)

C=1.5+3i

raiz=sqrt(C)

% Al escribir una expresión sin punto y coma al final
% MATLAB muestra su valor en pantalla.
```

Al correr el programa se obtienen como salida los siguientes resultados:

```
R =   -1.2341
raíz =  0 + 1.1109i
C =   1.5000 + 3.0000i
raíz =   1.5579 + 0.9628i
```

Comando ANGLE

Calcula el ángulo de fase (en radianes) de una matriz con elementos complejos. Si la matriz sólo tiene un elemento, calcula el ángulo de fase de ese complejo.

Sintaxis:

La sintaxis de la orden es:

Valor = angle(Matriz);

Valor es una matriz que almacena el valor del ángulo de fase del complejo (de 0 a 2*pi) que ocupa la misma posición en Matriz (el ángulo de fase del elemento 1,1 lo almacena en la posición 1,1).

Matriz es una matriz (puede tener un solo elemento) cualquiera con componentes complejas (los reales forman parte de los complejos).

Ejemplo simple de uso:

El siguiente ejemplo ilustra el uso de angle:

```
%Ejemplo de uso de angle.

C=[1 2i;1+3i 2.3+5i]

c=[1.5+3j]

disp('Para la matriz:');

v=angle(C)

disp('Para un complejo: (matriz de un solo elemento)');

v=angle(c)

% Al escribir una expresión sin punto y coma al final
% MATLAB muestra su valor en pantalla.
```

Al correr el programa se obtienen como salida los siguientes resultados:

```
C =        1.0000                  0 + 2.0000i
      1.0000 + 3.0000i   2.3000 + 5.0000i

c =   1.5000 + 3.0000i
```

Para la matriz:

```
v =        0      1.5708
      1.2490    1.1397
```

Para un complejo: (matriz de un solo elemento)

```
v =    1.1071
```

FICHA 13

SOFTWARE: MATLAB

TEMA: Integrales Definidas

INTEGRALES DEFINIDAS

Se resuelven utilizando el comando trapz

Comando TRAPZ

Calcula la integral definida entre dos límites de una función (área bajo la curva) representada por uno o dos vectores, como se explica más adelante. El cálculo de la integral se realiza numéricamente, por medio de una aproximación de la función a trapecios (en ningún momento calcula la integral simbólica).
Debido a que el cálculo de la integral es numérico, se deben construir vectores para calcular la integral. Por esta razón, es fundamental aclarar las características de los vectores, con el fin de tener un criterio para decidir como construir el vector de forma apropiada.

Sintaxis:

La sintaxis de la orden es:

Valor = trapz([Vector,] Matriz);

Los símbolos [ ] significan que Vector es opcional.
Matriz puede ser una matriz o un vector. Una matriz si se desea calcular la integral definida para varias funciones en el mismo rango (entre los mismos límites). Un vector si se desea calcular la integral para una sola función (su tamaño tiene relación con el tamaño de **Vector**, esta relación se muestra en detalle en la explicación de **Vector**).

Vector es el vector de los valores para los cuales se desea calcular la integral, tal que si Matriz es:

- *Un vector:* **Matriz** y **Vector** deben ser de la misma longitud (ya sean vectores fila, o columna). A cada valor almacenado en **Vector** corresponde el valor almacenado en **Matriz** (con el mismo subíndice).
- *Una matriz:* **Vector** debe ser un vector columna y **Matriz** tiene almacenadas las funciones por columnas (cada columna = una función), **Matriz** debe tener el mismo número de filas que **Vector**.

Si **Vector** se omite, es equivalente a introducir un vector con paso 1 (por ejemplo: [0, 1, 2, 3]), note que la integral no depende de los valores que se introducen en **Valor**, sino de su paso (ya que los valores de la función en cada punto están almacenados en **Matriz**), en otras palabras la integral sigue siendo la misma (en valor) si la corro hacia un lado y realizo la integral entre el nuevo par de límites.

Valor es donde se almacena el valor de la integral (un real si sólo se calculó para una función, y un vector fila si se calculó para varias).

Cómo construir vectores

La integral se realiza aproximando la curva (función) a una serie de rectas, con el fin de aproximar el área bajo la curva a una serie de trapecios contiguos. Por lo tanto la aproximación es buena si efectivamente la función se comporta como una recta (aproximadamente) en cada sub-intervalo, determinado por el paso y número de puntos que se tomen. Una forma empírica de verificar que los vectores están bien construidos es por medio de la orden plot, ya que esta función dibuja los vectores, aproximando la función de la misma forma que trapz. Por lo tanto, si al dibujar la curva con plot, esta se ve "suave", los vectores están bien definidos.

Ejemplo simple de uso:

El siguiente ejemplo ilustra el uso de trapz:

```
%Ejemplo de uso de trapz.

for i=1:100,

x(i, 1)=1+i/20; % Asina los valores de x entre 1 y 6 en incrementos de 0.05

y(i, 1)=x(i,1)+1;  % Define la función y=x+1

z(i, 1)=x(i,1)^2+1; % Define la función z=x^2+1

end;
```

```
% Los vectores x, y, z se definieron como vectores columna arriba, con
% el fin de demostrar el funcionamiento de trapz con varias funciones.
% estos vectores perfectamente hubieran podido ser fila, pero hubiera sido más
% difícil armar la matriz. Igualmente se requería construir un que x fuera columna.

A(:, 1)=y;

A(:, 2)=z;

integral=trapz(x, y)

integral=trapz(x, z) % Normalmente se usaría un nombre diferente al de arriba

integral=trapz(x, A) % Normalmente se usaría un nombre diferente al de arriba
% Al escribir una expresión sin punto y coma final MATLAB
% muestra en pantalla su valor.
```

Al correr el programa se obtiene la siguiente salida:

```
integral =   22.3988

integral =   76.5662

integral =
   22.3988    76.5662
```