UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES CON MENCIÓN EN GESTIÓN EMPRESARIAL DE TELECOMUNICACIONES

TEMA:

“Aplicación del programa MATLAB en la resolución de ecuaciones diferenciales aplicado a la materia de Cálculo Tres”

AUTOR:
Ramos Flores Santiago Ignacio

Tesis previa a la obtención del Título de:
INGENIERO EN TELECOMUNICACIONES CON MENCIÓN EN GESTIÓN EMPRESARIAL DE TELECOMUNICACIONES

TUTOR:
Ing. Córdova Rivadeneira Luis

Guayaquil, Ecuador
2012

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

CARRERA DE INGENIERÍA EN TELECOMUNICACIONES CON MENCIÓN EN GESTIÓN EMPRESARIAL DE TELECOMUNICACIONES

Certificamos que el presente trabajo fue realizado en su totalidad por el señor Santiago Ramos como requerimiento parcial para la obtención del Grado Académico de **Ingeniero en Telecomunicaciones con mención en Gestión Empresarial de Telecomunicaciones.**

____
Ing. Córdova Rivadeneira Luis
Director de tesis

____
Ing. Manuel Romero Paz
Decano de la Facultad Técnica

____
Ing. Armando Heras Sánchez
Director de Carrera

____
Ing. Luis Vallejo Samaniego
Coordinador Académico

____
Eco. Gladis Contreras Molina
Coordinadora Administrativa

____
Ing. Néstor Zamora C., MSc.
Revisor de Tesis

____
Ing. Efrén Herrera Muentes
Revisor de Tesis

Guayaquil, a los 10 del mes de diciembre del año 2012

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

CARRERA DE INGENIERÍA EN TELECOMUNICACIONES CON MENCIÓN EN GESTIÓN EMPRESARIAL DE TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Santiago Ramos**

DECLARO QUE:

La Tesis **"Aplicación del programa MATLAB en la resolución de ecuaciones diferenciales aplicado a la materia de Cálculo Tres"** previa a la obtención del **Grado Académico de Ingeniero en Telecomunicaciones con mención en Gestión Empresarial de Telecomunicaciones**, ha sido desarrollada en base a una investigación exhaustiva, respetando derechos intelectuales de terceros conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico de la tesis del Grado Académico en mención.

Guayaquil, a los 10 del mes de diciembre del año 2012

EL AUTOR

____

Santiago Ramos

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES CON MENCIÓN EN GESTIÓN EMPRESARIAL DE TELECOMUNICACIONES

AUTORIZACIÓN

Yo, **Santiago Ramos**

Autorizo a la Universidad Católica de Santiago de Guayaquil, la **publicación** en la biblioteca de la institución de la **Tesis de Ingeniería en Telecomunicaciones con mención en Gestión Empresarial de Telecomunicaciones** titulada: **"Aplicación del programa MATLAB en la resolución de ecuaciones diferenciales aplicado a la materia de Cálculo Tres"**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 10 del mes de diciembre del año 2012

EL AUTOR:

____

Santiago Ramos

AGRADECIMIENTO

En primer lugar, agradezco infinitamente a Dios, Ser Supremo, por permitirme lograr una meta más en mi vida.

Mi eterno agradecimiento a mis padres, quienes con su esfuerzo, dedicación y sabios consejos me han conducido por el camino del bien.

A mi familia y amigos, muchas gracias por su permanente apoyo.

SANTIAGO RAMOS

DEDICATORIA

El presente trabajo lo dedico de manera especial a mis padres, a su amor, dedicación y apoyo.

A mi familia por comprender y aceptar en silencio las horas de soledad que tuvieron que pasar, sacrificio que no fue en vano pues muy agradecido dedico a todos ellos el resultado en este escrito, y muy seguro estoy, servirá de ejemplo de esfuerzo y superación.

SANTIAGO RAMOS

TRIBUNAL DE SUSTENTACIÓN

ING. LUIS CÓRDOVA RIVADENEIRA
DIRECTOR DE TESIS

ING. NÉSTOR ZAMORA C., MSc.
REVISOR DE TESIS

ING. EFRÉN HERRERA MUENTES
REVISOR DE TESIS

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES CON MENCIÓN EN GESTIÓN EMPRESARIAL DE TELECOMUNICACIONES

CALIFICACIÓN

____

ING. LUIS CÓRDOVA RIVADENEIRA
DIRECTOR DE TESIS

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

El presente trabajo tiene como propósito ofrecer una orientación teórica comprensible sobre Ecuaciones Diferenciales y su implementación en el software MATLAB, estableciendo una comparación entre la resolución usual de las EDOs, es decir, la resolución utilizando Álgebra y Cálculo; y, la resolución operando los comandos del programa de cálculo técnico científico MATLAB.

Para hacer posible dicha intención se presenta una revisión teórica del tema y se procede a implementarlo en este software, tanto algebraico como numérico, mostrando y comparando alternativamente las resoluciones numéricas contra las algebraicas, evaluando la eficiencia del programa.

Está dirigido a los alumnos que cursan la materia Cálculo Tres en las distintas carreras de Ingeniería y a cualquier persona interesada en el tema, puesto que servirá como material de apoyo para el aprendizaje de ecuaciones diferenciales ordinarias. El alumno estará en la capacidad de representar la solución numérica de varios problemas y adquirir las nociones y herramientas suficientes para resolver otros problemas más complicados.

ABSTRACT

This research aims provide a comprehensive theoretical orientation on Differential Equations and its implementation in the MATLAB software, drawing a comparison between the usual resolution of ODEs, namely, solving using algebra and calculus; and, operating resolution program commands MATLAB technical computing science.

To enable this intention presents a theoretical review of the topic and proceed to deploy this software, both algebraic and numerical, alternately displaying and comparing numeric resolutions against algebraic, evaluating program efficiency.

It is aimed at students taking the matter Calculus Three in different engineering careers and to anyone interested in the subject, since it serve as support material learning of ordinary differential equations. The student will be able to represent the numerical solution of several problems and acquire the necessary tools and notions to solve other problems more complicated.

CAPÍTULO 1

1. INTRODUCCIÓN

Lo que se pretende con esta tesis es dar a conocer las diferentes aplicaciones que tiene el programa MATLAB en la resolución de Ecuaciones Diferenciales aplicado a la Materia de Cálculo Tres. Se hizo este proyecto de tesis porque es necesario que los estudiantes tengan a la mano herramientas que les ayude a resolver ejercicios matemáticos, *MATLAB* ofrece estas ventajas ya que se caracteriza por ser un software orientado al cálculo numérico científico e ingeniería, integra cálculo numérico, computación de matrices y gráficos en un entorno de trabajo cómodo para el usuario. Su nombre significa Laboratorio de Matrices y fue escrito inicialmente en base a los ya existentes paquetes de cálculo matricial LINPAK y EISPACK, con el pasar de los tiempos se han agregado paquetes o libretas denominadas *Toolboxes* que han sido desarrolladas para diferentes áreas científicas.

Adicionalmente, se realizó con el objetivo y propósito de diferenciar o contrastar entre la resolución usual de ecuaciones diferenciales ordinarias, es decir, la resolución normal usando y empleando el Álgebra y el Cálculo con la resolución operando los comandos de programa de Cálculo Técnico Científico de MATLAB.

Lo que comprende esta tesis es llegar inicialmente a expresar una ecuación y a realizar su solución normal empleando los conocimientos obtenidos en la materia de Cálculo Tres utilizando el álgebra y el cálculo, luego obtenemos la solución general y particular utilizando los comandos de *MATLAB,* así mismo se representa gráficamente las soluciones.

Esta Tesis está dirigida al lector interesado en el tema y, fundamentalmente y sobre todo, a los alumnos que cursan la asignatura de Cálculo Tres en las diversas carreras de Ingeniería que configura la oferta Académica de la Facultad de Educación Técnica para el Desarrollo de la Universidad Católica de Santiago de Guayaquil.

1.1. PLANTEAMIENTO DEL PROBLEMA.-

Actualmente, el software de MATLAB a pesar de ser una herramienta muy utilizada en las Universidades locales e internacionales, en la Facultad de Educación Técnica para el Desarrollo de la Universidad Católica de Santiago de Guayaquil poco se la utiliza especialmente en los semestres de básica, y como consecuencia el estudiante llega a los semestres más avanzados encontrándose con la sorpresa de no poder modelar un sistema dinámico de control en otras materias donde el uso de MATLAB es importante, como Teoría de control uno y dos. Entonces, se ratifica la problemática, que es el poco conocimiento de los estudiantes de esta herramienta y de las bondades que ofrece no sólo en el ámbito matemático sino también en la simulación y modelación.

1.1.1. DELIMITACIÓN DEL PROBLEMA.-

Con lo ya expuesto, esta tesis está delimitada para la modelación, resolución y graficación de ecuaciones y sistemas de ecuaciones diferenciales utilizando MATLAB, con el propósito de exponer la sencillez y rapidez en el aprendizaje y utilización de esta herramienta en la materia Cálculo, tanto a nivel básico como avanzado.

1.1.2. JUSTIFICACION.-

A través del tiempo MATLAB ha evolucionado vertiginosamente y ha podido crecer gracias a las aportaciones de muchos usuarios. En entornos Universitarios del País, esta herramienta de software se ha convertido en un material básico de instrucción, tanto para cursos de matemáticas aplicadas como para cursos muy avanzados en otras áreas. En entornos industriales se utiliza para la investigación y resolución de problemas prácticos y cálculos de ingeniería. Aplicaciones típicas del MATLAB son el cálculo numérico, la realización de algoritmos, la resolución de problemas con formulación matricial, la estadística, la optimización, y así muchas otras aplicaciones en el contexto de las matemáticas, y como no destacar al *Software Matlab* para la aplicación en el estudio, simulación y diseño de los sistemas dinámicos y de control.

1.2. ANTECEDENTES.-

La ciencia de las matemáticas, inventada hace mucho tiempo atrás por los griegos, ha estado hasta hoy en día presente en el universo, tanto así que actualmente el hombre se sigue valiendo de ella para conocer los diferentes fenómenos o sucesos que pasan o nos rodean para cumplir los objetivos de descubrimiento.

El avance tecnológico en las computadoras conjuntamente con el desarrollo de programas, y la actual ciencia de la computación conjuntamente con las matemáticas fundamentales y sus aplicaciones, son campos muy intensos pero con una gran importancia.

Los diferentes programas de matemáticas que interactúan con la computadora en sus diferentes aplicaciones son muy amplios, sobre todo para la resolución de problemas numéricos muy complicados cuya resolución manual algebraica hubiera sido prácticamente imposible de realizar hace unos años atrás. Por estos motivos, en esta tesis se incentiva al estudiante a conocer y a utilizar herramientas que le ayuden a la realización de experimentos en muchas ciencias, desde las propias matemáticas hasta las sociales, con costos y esfuerzo mínimos utilizando herramientas tecnológicas.

Adicionalmente, la Modelación Matemática se fundamenta en otras distintas ramas de la matemática, como lo son: las Ecuaciones lineales, vectores, análisis funcional, Ecuaciones diferenciales, y otras disciplinas de las cuales se requieren conceptos y conocimientos.

En esta tesis se analiza la Teoría de las Ecuaciones Diferenciales por ser unas de las áreas de las matemáticas más extensa e importante y principalmente fundamental en la Modelación, en especial para la carrera Técnica en Ingeniería que ofrece esta facultad, con una ecuación Diferencial se pueden modelar distintos fenómenos físicos, del cual hay varios métodos de solución para algunas de estas ecuaciones, pero es importante recalcar que no todas las soluciones analíticas son funciones fáciles de resolver, sin embargo, solucionan un gran número de problemas especialmente si utilizamos herramientas tecnológicas.

1.3. OBJETIVOS.-

Los objetivos planteados para este proyecto de investigación son los siguientes:

1.3.1. OBJETIVO GENERAL.-

Fortalecer el estudio, la comprensión y aplicación del Programa Matlab en la resolución de Ecuaciones Diferenciales aplicado a la Materia de Cálculo Tres.

1.3.2. OBJETIVOS ESPECIFICOS.-

- Estudiar los diferentes temas aplicados en la materia de Cálculo Tres con respecto a las ecuaciones diferenciales.
- Demostrar la solución manual algebraica de los diferentes ejercicios de los temas tratados de acuerdo a lo aprendido en la materia de Cálculo Tres.
- Estudiar y demostrar el manejo y las funciones básicas del *Matlab* y los respectivos comandos para la resolución de los diferentes ejercicios aplicados a las ecuaciones diferenciales.

1.4. HIPÓTESIS.-

Comprobar lo aprendido teóricamente con las debidas prácticas de laboratorio y corroborar que se cumplen las distintas definiciones. Posterior a verificar lo anteriormente expresado, el presente escrito podría ser aprovechado como orientación, soporte y refuerzo de las nociones adquiridas para alumnos y catedráticos de la profesión.

Introducción a las Ecuaciones diferenciales ordinarias

8.1 Primeras definiciones.
8.2 Solución de una Ecuación Diferencial

Ecuaciones Diferenciales de primer orden

8.3 Ecuaciones diferenciales y modelos matemáticos
8.4 Integrales como soluciones generales y particulares
8.5 Campos direccionales y curvas solución
8.6 Ecuaciones separables y aplicación
8.7 Ecuaciones lineales
8.8 Métodos de sustitución y ecuaciones exactas.

Introducción al sistema de ecuaciones diferenciales

8.9 Sistemas de primer orden y aplicaciones
8.10 El método de eliminación
8.11 Métodos numéricos para sistemas

Introducción a Matlab

8.12 Generalidades
8.13 Características de la ventana de comandos
8.14 Cómo utilizar el *workspace*
8.15 Formatos de números
8.16 Gestión de directorios
8.17 M-files.
8.18 Operaciones con vectores
8.19 Representaciones gráficas.
8.20 Toolbox de control
8.13 Aritmética elemental
8.14 Almacenando resultados en variable
8.15 Precisión y formato de los resultados
8.16 Construcción de Matrices y operaciones elementales
8.17 Funciones de Matlab
8.18 Cálculos Básicos

Cálculos numéricos en Matlab

8.19 Polinomios
8.20 Resolución de Ecuaciones
8.21 Derivación numérica
8.22 Máximos y mínimos
8.23 Cálculos en algebre lineal
8.24 Resolución de Ecuaciones Diferenciales
8.25 Ecuaciones diferenciales de primer orden
8.26 Sistemas de Ecuaciones diferenciales de primer orden
8.27 Ecuaciones diferenciales de orden superior al primero

CAPÍTULO 2

2. INTRODUCCIÓN A LAS ECUACIONES GENERALES ORDINARIAS

2.1. PRIMERAS DEFINICIONES.-

Una **ecuación diferencial** es una ecuación en la que intervienen derivadas de una o más funciones desconocidas. Dependiendo del número de variables independientes respecto de las que se deriva, las ecuaciones diferenciales se dividen en:

- **Ecuaciones diferenciales ordinarias**, aquellas que contienen derivadas respecto a una sola variable independiente.

- **Ecuaciones en derivadas parciales**, aquellas que contienen derivadas parciales respecto a dos o más variables independientes.

2.1.1. ECUACIÓN DIFERENCIAL ORDINARIA.-

En matemáticas, una **ecuación diferencial ordinaria** ("EDO") es una relación que contiene funciones de una sola variable independiente, y una o más de sus derivadas con respecto a esa variable.

Si *F* es una relación o función, la ecuación diferencial ordinaria es una función implícita *(y = y(x))*.

$$F(x, y, y', y'', \ldots., y^{(n)}) = 0 \qquad (1a)$$

Las ecuaciones diferenciales ordinarias son importantes en múltiples campos de estudio como la geometría, mecánica y astronomía, además de muchas otras aplicaciones.

Se ha estudiado mucho sobre la resolución de este tipo de ecuaciones, y la teoría para ecuaciones lineales está desarrollada en casi su totalidad. Mas, la mayoría de las ecuaciones diferenciales son no-lineales, y gran parte de ellas no tienen solución exacta.

La ecuación diferencial lineal general de orden n está dada por:

$$a_n(t)y^{(n)} + a_{n-1}(t)y^{(n-1)} + \ldots + a_1(t)y' + a_0(t)y = g(t) \qquad \text{(1b)}$$

Donde los a_i representan funciones dependientes de t.

Una solución de la ecuación (1a) o (1b) será una familia de curvas o funciones del tipo $y = f(t)$ que substituida dentro de la ecuación, la convierte en una igualdad en la que todos los términos son conocidos.

Si y es una función desconocida:

$$y : \mathbb{R} \rightarrow \mathbb{R}$$

de x siendo $y^{(n)}$ la enésima derivada de y, entonces una ecuación de la forma

$$F(x, y, y', \ldots., y^{(n-1)}) = y^{(n)} \qquad (1)$$

es llamada una **ecuación diferencial ordinaria** (EDO) de **orden** n. Para funciones vectoriales,

$$y : \mathbb{R} \to \mathbb{R}^m,$$

la ecuación (1) es llamada un **sistema de ecuaciones lineales diferenciales** de dimensión *m*.

Cuando una ecuación diferencial de orden *n* tiene la forma

$$F(x, y, y', y'', \ldots, y^{(n)}) = 0$$

es llamada una ecuación diferencial **implícita**, mientras que en la forma

$$F(x, y, y', y'', \ldots, y^{(n-1)}) = y^{(n)}$$

es llamada una ecuación diferencial **explícita**.

Una ecuación diferencial que no depende de *x* es denominada **autónoma**.

Se dice que una ecuación diferencial es **lineal** si *F* puede ser escrita como una combinación lineal de las derivadas de *y*

$$y^{(n)} = \sum_{i=0}^{n-1} a_i(x) y^{(i)} + r(x)$$

siendo, tanto $a_i(x)$ como *r(x)* funciones continuas de *x*. La función *r(x)* es llamada el **término fuente** (*source term*); si *r(x)*=0 la ecuación diferencial lineal es llamada **homogénea**, de lo contrario es llamada **no homogénea**.

2.2. SOLUCIÓN DE UNA ECUACIÓN DIFERENCIAL.-

Dada una ecuación diferencial

$$F(x, y, y', \ldots., y^{(n)}) = 0$$

una función *u: /*□ **R** → **R** es llamada la **solución** o curva integral de *F*, si *u* es *n* veces derivable en /, y

$$F(x, u, u', \ldots., u^{(n)}) = 0 \quad x \in I$$

Dadas dos soluciones *u: J* □ **R** → **R** y *v: /*□ **R** → **R**, *u* es llamada una **extensión** de *v* si / □ *J*, y

$$u(x) = v(x) \quad x \in I$$

En pocas palabras, una función *f* o *f(x)* es una solución de una ecuación diferencial si al sustituir *y* por *f(x)* se obtiene una identidad para todo *x* en un intervalo.

2.2.1. SOLUCIÓN GENERAL.-

Una solución que no tiene extensión es llamada una **solución general.**

Se llama **solución general** de una ecuación diferencial a toda relación entre las variables, libres de derivadas, que satisface dicha ecuación diferencial.

Por lo común, la solución general de una ecuación diferencial de orden *n* contiene *n* variables arbitrarias correspondientes a *n* constantes de integración. Integrar o resolver una ecuación diferencial es hallar su solución general.

2.2.2. SOLUCIÓN PARTICULAR.-

Una **solución particular** es derivada de la solución general mediante la fijación de valores particulares para las constantes, a menudo elegidas para cumplir *condiciones iniciales*.

2.2.3. SOLUCIÓN SINGULAR.-

Una **solución singular** es una solución que no puede ser derivada de la solución general.

2.3. ECUACIONES DIFERENCIALES DE PRIMER ORDEN.-

Una ecuación diferencial de primer orden con la condición inicial se expresa de la siguiente forma:

$$[L] = \begin{cases} \dfrac{dy}{dt} = f(t, y) \\ y(t_0) = y_0 \end{cases}$$

Donde $y(t_0) = y_0$ es la condición inicial.

2.3.1. ECUACIONES DIFERENCIALES Y MODELOS MATEMÁTICOS.-

Las leyes del universo están escritas en el lenguaje de las matemáticas. El álgebra es suficiente para resolver muchos problemas estáticos, pero los fenómenos naturales más interesantes implican cambios y se describen sólo por medio de ecuaciones que relacionen las cantidades que cambian.

Puesto que la derivada $dx/dt=f'(t)$ de la función f es la razón a la cual la cantidad $x=f(t)$ está cambiando respecto a la variable independiente t, es natural que las ecuaciones que incluyan derivadas se usen con frecuencia para describir el universo cambiante. Una ecuación que relaciona una función desconocida y una o más de sus derivadas se llama **ecuación diferencial**.

El estudio de las ecuaciones diferenciales tiene tres objetivos principales:

1. Descubrir la ecuación diferencial que describa una situación física especificada;
2. Determinar –ya sea de manera exacta o aproximada- la solución apropiada de esa ecuación;
3. Interpretar la solución que se encuentre.

En álgebra, por lo general se buscan los *números* desconocidos que satisfagan una ecuación, tal como $x^3+7x^2-11x+41=0$. Por el contrario, al resolver una ecuación diferencial, nos enfrentamos a buscar *funciones* desconocidas $y=y(x)$ para las cuales una identidad como $y'(x)=2xy(x)$, esto es, que la ecuación diferencial

$$\frac{dy}{dx} = 2xy$$

se cumpla en algún intervalo de números reales. Por lo general, si es posible, se necesitará determinar *todas* las soluciones de la ecuación diferencial.

2.3.1.1. MODELOS MATEMÁTICOS.-

Un modelo matemático es la representación de todas las características importantes de un sistema con el propósito de derivar las ecuaciones matemáticas que determinen su comportamiento. Debe incluir los mínimos detalles del sistema, tal que dicho comportamiento pueda ser representado por una ecuación. Puede ser lineal o no lineal. Un modelo matemático permite soluciones rápidas y simples, sin embargo los modelos no lineales, revelan algunas veces ciertas características del sistema que los modelos lineales no proporcionan.

La descripción matemática de un sistema o fenómeno se llama modelo matemático. Con frecuencia podemos describir el comportamiento de algún sistema o ecosistema o cualquier fenómeno de la vida real en términos matemáticos; dicho sistema puede ser físico, sociológico o hasta económico.

La formulación de un modelo matemático de un sistema implica:

1. Identificar las variables causantes del cambio del sistema estableciendo una notación matemática.
2. Formular el modelo. Establecer un conjunto de hipótesis razonables que incluya todas las leyes empíricas aplicables al sistema. En algunas ocasiones, basta contar con modelos de baja resolución. Dado que las hipótesis acerca de un sistema implican con frecuencia

la razón, o tasa, de cambio de una o más de las variables, el enunciado matemático de todas esas hipótesis es una o más ecuaciones donde intervienen derivadas. En otras palabras, el modelo matemático puede ser una ecuación o un sistema de ecuaciones diferenciales.

3. Resolver el modelo planteado.
4. Interpretar la conclusión matemática. Se dice que el modelo es razonable si su solución es consistente con los datos experimentales o los hechos conocidos acerca del comportamiento del sistema. Si las predicciones que se basan en la solución son deficientes, podemos aumentar el nivel de resolución del modelo o elaborar hipótesis alternativas sobre los mecanismos de cambio del sistema; entonces, se repiten los pasos del proceso de modelado.
5. Al aumentar la resolución, se aumenta la complejidad del modelo matemático y la probabilidad de que se deba conformar con una solución aproximada.

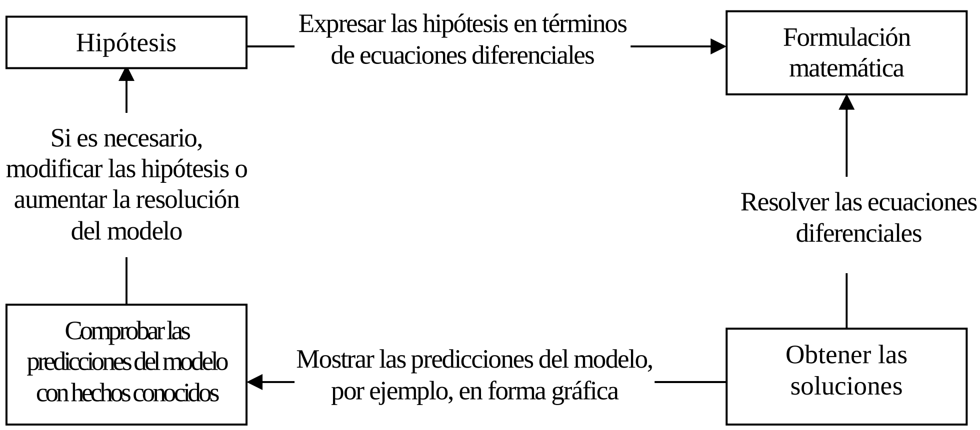


Figura 1.- Formulación de modelo matemático.
Fuente: http://es.scribd.com/doc/57741542/5/I-2-2-Soluciones-generales-y-particulares-de-una-E-D

El proceso crucial de la *modelación matemática* incluye:

1. Formulación de un problema del mundo real en términos matemáticos; esto es, la construcción de un modelo matemático.

2. Análisis o solución del problema matemático resultante.
3. Interpretación de los resultados matemáticos en el contexto original de la situación del mundo real; por ejemplo, responder la pregunta originalmente planteada.

Un **modelo matemático** consiste en una lista de variables (*P* y *t*) que describe la situación dada, junto con una o más ecuaciones que relacione estas variables ($dP/dt=kP$, $P(0)=P_0$) que son conocidas o se supone son válidas. El análisis matemático consiste en la solución de estas ecuaciones (para *P* como una función de *t*). Por último, aplicamos estos resultados matemáticos para intentar responder la pregunta original del mundo real.

Sin embargo, es probable que ninguna solución de la ecuación diferencial cumpla *toda* la información conocida. En tal caso se debe sospechar que dicha ecuación diferencial podría no describir de manera adecuada al mundo real.

Un modelo matemático satisfactorio debe estar sujeto a dos requisitos contradictorios: debe ser suficientemente detallado para representar la situación del mundo real con relativa exactitud; y, aún así, debe ser lo suficientemente sencillo para hacer práctico el análisis matemático. Si el modelo es demasiado detallado que represente por completo la situación física, entonces el análisis matemático puede ser demasiado difícil de llevar a cabo. Si el modelo es demasiado simple, los resultados pueden ser tan imprecisos que sean inútiles. Por lo tanto, hay un compromiso ineludible entre lo que es físicamente realista y lo que es matemáticamente posible. Es así, que el paso más crucial y delicado en el proceso es la construcción de un modelo que salve adecuadamente esta brecha entre el realismo y la factibilidad. Se deben encontrar caminos para simplificar el modelo matemáticamente sin sacrificar las características esenciales de la situación del mundo real.

A continuación se muestran algunos ejemplos que ilustran el proceso de traducir leyes y principios científicos en ecuaciones diferenciales. En cada uno de los ejemplos la variable independiente es el tiempo *t*, pero habrá otros casos en los que una cantidad diferente del tiempo será la variable independiente.

EJEMPLO 1.-

La ley de enfriamiento de Newton puede ser establecida de la siguiente forma: La *tasa de cambio* de la temperatura *T(t)* de un cuerpo con respecto al tiempo *t* es proporcional a la diferencia entre *T* y la temperatura *A* del medio ambiente.

$$\frac{dT}{dt} = -k(T - A)$$

donde *k* es una constante positiva. Se observa que si $T > A$, entonces $dT/dt < 0$, de modo que la temperatura es una función decreciente de *t* y el cuerpo se está enfriando. Pero, si $T < A$, entonces $dT/dt > 0$, y por tanto *T* está aumentando.

Así, la ley física se traduce a una ecuación diferencial. Si se dan valores a *k* y *A*, se puede determinar una fórmula explícita para *T(t)*, y entonces -con la ayuda de esta fórmula- se podrá predecir la temperatura futura del cuerpo.

EJEMPLO 2.-

La *tasa de cambio con respecto al tiempo* de una población *P(t)* con tasas de natalidad y mortalidad constantes es, en muchos casos sencillos, proporcional al tamaño de la población. Esto es,

$$\frac{dP}{dt} = kP \qquad \text{(a)}$$

donde *k* es la constante de proporcionalidad.

Analizando con más detalle el ejemplo anterior. Cada función de la forma

$$P(t) = Ce^{kt} \qquad \text{(b)}$$

es una solución de la ecuación diferencial

$$\frac{dP}{dt} = kP$$

en (a). Se verifica esta afirmación de la siguiente forma:

$$P'(t) = Cke^{kt} = k\,(Ce^{kt}) = kP(t)$$

para todos los números reales *t*. Puesto que cada función de la forma dada en (b) sustituida en la ecuación (a) produce una identidad, todas esas ecuaciones son soluciones de la ecuación (a).

Por lo tanto, aún si el valor de la constante *k* es conocido, la ecuación diferencial *dP/dt=kP* tiene un *número infinito* de soluciones diferentes de la forma $P(t)=Ce^{kt}$, uno para cada elección de la constante “arbitraria” *C*. Esto es común de las ecuaciones diferenciales. También es afortunado, ya que puede permitir utilizar información adicional para seleccionar de entre todas estas soluciones una particular que se adecue a la situación estudiada.

2.3.2. INTEGRALES COMO SOLUCIONES GENERALES Y PARTICULARES.-

En una ecuación diferencial, la incógnita no es un número, sino una función del tipo *y = F(x).*

Hallar todas las funciones que satisfacen una determinada ecuación diferencial, significa resolver la misma. Todas estas funciones que la satisfacen reciben el nombre de **SOLUCIONES** o **INTEGRALES**. Toda ecuación diferencial admite, en general, infinitas soluciones, cuyas gráficas se llaman **CURVAS INTEGRALES**.

Por Ejemplo: Sea la ecuación

$$dy/dx = f(x) \Rightarrow dy = f(x)\,dx$$

integrando

$$y = y(x) \Rightarrow \int dy = \int f(x)\,dx + C$$

donde C es una constante arbitraria.

Esta solución representa la ecuación de una *familia* o *haz de curvas*, cada una de las cuales puede determinarse fijando el correspondiente valor de C. Es decir, por cada punto del plano en que $y = f(x)$ cumple ciertas condiciones impuestas, pasa una, y solamente una curva que satisface a la ecuación diferencial.

Toda expresión que satisface a la ecuación diferencial de primer orden, cualquiera sea el valor de la constante C se llama **SOLUCIÓN GENERAL** de la ecuación. Es decir, es una familia de soluciones con un parámetro.

Si $G(x)$ es una antiderivada particular de f -es decir, si $G'(x) \equiv f(x)$- entonces,

$$y(x) = G(x) + C$$

Las gráficas de cualesquiera dos soluciones, tales como $y_1(x) = G(x) + C_1$ y $y_2(x) = G(x) + C_2$ con el mismo intervalo *I* son paralelas en el sentido ilustrado en las figuras 2 y 3. En ellas se puede observar que la constante *C* es la distancia vertical entre las dos curvas $y(x) = G(x)$ y $y(x) = G(x) + C$.

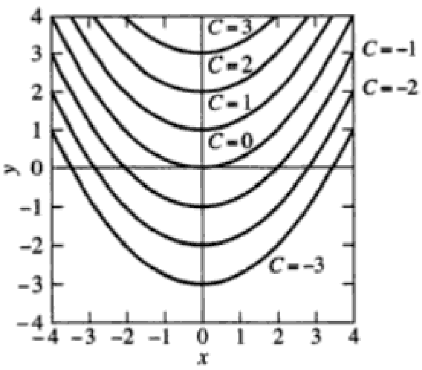


Figura 2.- Gráficas de $y = x^2 + C$ para diferentes valores de *C.*

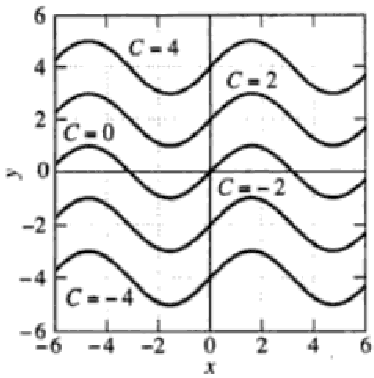


Figura 3.- Gráficas de $y = Sen\ x + C$ para diferentes valores de *C.*

Fuente: EDWARDS, C. Henry; PENNEY, David E. *Ecuaciones diferenciales.*

Si fijado cualquier punto $P(x_0, y_0)$ por el que debe pasar necesariamente la solución de la ecuación diferencial, existe un único valor de *C*, y por lo tanto de la curva integral correspondiente, que satisface la ecuación inicial, esta recibirá el nombre de **SOLUCIÓN PARTICULAR** de la ecuación. En este caso el punto $P(x_0, y_0)$ recibe el nombre de **CONDICIÓN INICIAL** y supone el conocimiento previo de un punto de la solución, que generalmente se obtiene experimentalmente.

$$\frac{dy}{dx} = f(x), \quad y(x_0) = y_0$$

Es decir, una solución de una ecuación diferencial que no tiene parámetros arbitrarios es una **solución particular**.

Lo enunciado anteriormente corresponde al procedimiento tradicional para la solución de ecuaciones diferenciales de primer orden. Generalmente, primero se determina una *solución general* que incluya una constante arbitraria C. Posteriormente, se intenta obtener, mediante una apropiada elección de C, una *solución particular* que satisfaga la condición inicial dada $y(x_0) = y_0$.

2.3.3. CAMPOS DIRECCIONALES Y CURVAS SOLUCIÓN.-

Analizando la ecuación diferencial de primer orden $dy/dx = y$, ésta significa que las pendientes de las tangentes a la gráfica de una solución están determinadas por la función $f(x, y) = y$.

Cuando $f(x, y)$ se mantiene constante -esto es, cuando y es igual a C, donde C es cualquier constante real- se obliga a que la pendiente de las tangentes a las curvas de solución tenga el mismo valor constante a lo largo de una línea horizontal; por ejemplo, para $y = 2$ se puede trazar una serie de segmentos lineales cortos o **elementos lineales** (cada uno de pendiente 2) con su punto medio en la línea. Como se puede apreciar en la figura 4, las curvas de solución cruzan esta recta horizontal en cada punto tangente a los elementos lineales.

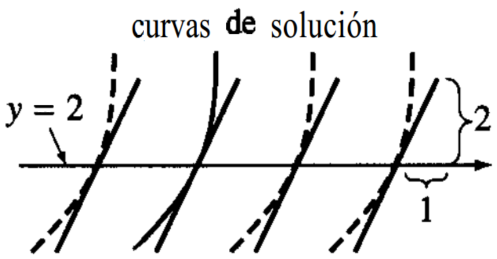


Figura 4. Curvas de solución cruzan recta horizontal en cada punto tangente a elementos lineales.
Fuente: http://mazinger.sisib.uchile.cl/repositorio/ap/ciencias_quimicas_y_farmaceuticas/apmat4d/09a.html

Si se evalúa f de forma sistemática en una red de puntos rectangular en el plano xy y se traza un elemento lineal en cada punto (x, y) de la red con pendiente $f(x, y)$, entonces la colección de estos elementos lineales se llama **campos de dirección** o **campos de pendiente**, de la ecuación diferencial $dy/dx = f(x, y)$.

La ecuación *y* = *C* representa una familia a un parámetro de líneas horizontales. En general, cualquier miembro de la familia *f(x, y)* = *C* se llama **isoclina**, que literalmente significa curva a lo largo de la cual la inclinación de las tangentes es igual. Cuando se hace variar el parámetro *C*, se obtiene un conjunto de isoclinas en que los elementos lineales se construyen adecuadamente. La totalidad de esos elementos lineales se llama de diversos modos: *campo de direcciones*, ***campo direccional***, *campo de pendientes* o *campo de elementos lineales* de la ecuación diferencial *dy/dx* = *f(x, y)*.

Como se puede apreciar en la figura 5 a), el campo de direcciones recuerda las "líneas de flujo" de la familia de curvas de solución de la ecuación diferencial *y'* = *y*. Si se requiere una solución que pase por el punto (0, 1), se debe formar una curva, como se indica (en celeste) en la figura 5 b), que pase por este punto de modo que atraviese las isoclinas con las inclinaciones adecuadas.

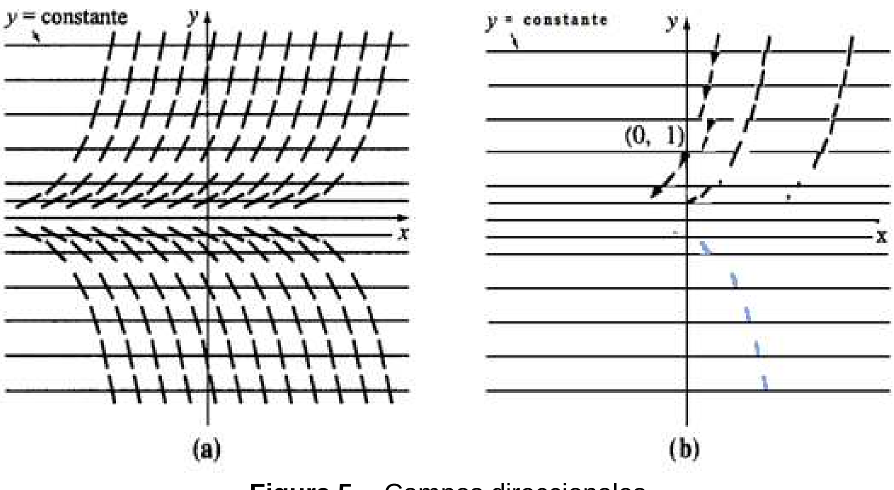


Figura 5.- Campos direccionales.
Fuente: http://mazinger.sisib.uchile.cl/repositorio/ap/ciencias_quimicas_y_farmaceuticas/apmat4d/09a.html

En algunas ocasiones la gráfica de una solución de una ecuación diferencial se denomina **curva solución** de la ecuación. Desde este punto de vista geométrico, una curva solución de una ecuación diferencial es una curva en el plano cuya recta tangente en cada punto *(x, y)* tiene pendiente *m = f(x, y)*.

Se puede decir entonces que, el campo direccional es un bosquejo con pequeños segmentos de recta trazados en un sistema de coordenadas cartesianas *(x, y)*, donde se muestra el comportamiento de la pendiente (derivada) que le corresponde a la **curva solución**.

2.3.4. ECUACIONES SEPARABLES Y APLICACIONES.-

Es el tipo más sencillo de las ecuaciones diferenciales de primer orden. Estas son ecuaciones que incluyen sólo a la primera derivada de la función desconocida y son tales que pueden separarse, una en cada lado de la ecuación.

Así, la ecuación diferencial de primer orden

$$\frac{dy}{dx} = H(x,y) \qquad (1)$$

se denomina **separable** a condición de que *H(x, y)* pueda escribirse como el producto de una función de *x* y una función de *y*:

$$\frac{dy}{dx} = g(x)\phi(y) = \frac{g(x)}{f(y)}$$

donde *ϕ(y) = 1/f(x)*. En este caso, las variables *x* y *y* pueden separarse -aislarse en miembros opuestos de una ecuación- escribiendo de manera informal la ecuación

$$f(y)dy = g(x)dx$$

se entiende será una anotación concisa para la ecuación diferencial

$$f(y)\frac{dy}{dx} = g(x) \qquad (2)$$

Es fácil resolver este tipo especial de ecuaciones diferenciales con sólo integrar ambos miembros con respecto a *x*:

$$\int f(y(x))\frac{dy}{dx}dx = \int g(x)dx + C$$

o de forma equivalente,

$$\int f(y)dy = \int g(x)dx + C \qquad (3)$$

Todo lo que se necesita es que puedan determinarse las antiderivadas

$$F(y) = \int f(y)dy \quad \text{y} \quad G(x) = \int g(x)dx$$

Para comprobar que las ecuaciones (2) y (3) son equivalentes, hay que observar la siguiente consecuencia de la regla de la cadena:

$$D_x[F(y(x))] = F'(y(x))y'(x) = f(y)\frac{dy}{dx} = g(x) = D_x[G(x)]$$

que a su vez es equivalente a

$$F(y(x)) = G(x) + C \qquad (4)$$

ya que dos funciones tienen la misma derivada en un intervalo si, y sólo si, ellas sólo difieren por una constante en ese intervalo.

En un principio, las ecuaciones diferenciales permiten resolver problemas geométricos que son difícilmente abordables desde otro punto de partida.

EJEMPLO 1.-

Por ejemplo, se considera el siguiente: determinar la familia de curvas del plano que verifican que en cada punto P su recta normal corta al eje Y en un punto M tal que la distancia de M a P es uno.

Para resolver este problema, supongamos que $y(x)$ es una curva de la familia y se va a determinar la ecuación diferencial de dicha familia. Resolviendo la ecuación diferencial se obtendrá la familia uniparamétrica de curvas que cumplen con la condición pedida. Se fija un punto arbitrario (x, y) de la gráfica de $y(x)$ y sea $Y - y = y'(X - x)$ su recta tangente y por tanto $Y - y = -1/y'(X - x)$ (nótese que las variables de las rectas se las escribe en mayúsculas). El punto de corte con el eje Y se lo obtiene resolviendo el sistema

$$\begin{cases} Y - y = -\frac{1}{y'}(X - x) \\ Y = 0 \end{cases}$$

Despejando X se tiene que el punto de corte es

$M = (x + yy', 0)$

Ahora se calcula la distancia de M a P,

$$d(P, M) = \sqrt{(x - yy' - x)^2 + (0 - y)^2} = \sqrt{(yy')^2 + y^2} = \mathbf{1}$$

de donde

$$(yy')^2 + y^2 = 1$$

Se despeja y', teniendo las ecuaciones de variables separables

$$y' = \pm\frac{1}{y\sqrt{1 - y^2}}$$

que tienen por solución, en el caso positivo

$$-\frac{1}{3}\sqrt{(1 + y^2)^3} = x + c$$

y en el caso negativo

$$\frac{1}{3}\sqrt{(1 + y^2)^3} = x + c$$

EJEMPLO 2.-

La aceleración de un objeto que se mueve a lo largo de un eje coordenado está dada por $a(t) = (2t + 3)^{-3}$ en metros por segundo. Si la velocidad en $t = 0$ es 4 metros por segundo, encontrar la velocidad 2 segundos más tarde.

Solución:

Se empieza con la ecuación diferencial

$$\frac{dv}{dt} = (2t + 3)^{-3}$$

Se realiza la integración y queda

$$v = \int (2t + 3)^{-3}\, dt = \frac{1}{2}\int (2t + 3)^{-3}\, 2dt = \frac{1}{2}\frac{(2t + 3)^{-2}}{-2} + c = -\frac{1}{4(2t + 3)^2} + c$$

Como $v = 4$ en $t = 0$

$$4 = -\frac{1}{4(3)^2} + c$$

Da $c = \frac{145}{36}$. Así,

$$v = -\frac{1}{4(2t + 3)^2} + \frac{145}{36}$$

En $t = 0$

$$v = -\frac{1}{4(49)} + \frac{145}{36} = 4.023 \; metros \; por \; segundo$$

2.3.5. ECUACIONES LINEALES.-

Una ecuación diferencial ordinaria de primer orden que es lineal con respecto a la función incógnita (variable dependiente) y su derivada se denomina una *ecuación diferencial lineal de primer orden*.

Según la definición, la ecuación diferencial ordinaria lineal de primer orden tiene la forma general:

$$a_0(x)\frac{dy}{dx} + a_1(x)y = q(x)$$

Si se divide ambas partes de esta ecuación por el coeficiente $a_0(x)$ se obtiene la nueva forma general para una ecuación lineal de primer orden:

$$\frac{dy}{dx} + p(x)y = g(x)$$

Donde las funciones $p(x) = a_1(x)/a_0(x)$ y $g(x) = q(x)/a_0(x)$.

La linealidad significa que todos los coeficientes $a_0(x)$, $a_1(x)$, $p(x)$ y las funciones $q(x)$, $g(x)$ son funciones sólo de x y que la función incógnita $y(x)$ y su derivada $y'(x)$ están en su primera potencia. Existen algunos métodos que permiten resolver cualquier ecuación lineal de primer orden.

Como ejemplo, para resolver la ecuación

$$\frac{dy}{dx} = 2xy \quad (y > 0) \qquad (1)$$

se multiplica ambos miembros por el factor *1!y* para obtener

$$\frac{1}{y} \cdot \frac{dy}{dx} = 2x \Rightarrow D_x(\ln y) = D_x\,(x^2) \qquad (2)$$

Ya que cada miembro de la ecuación en (2) es reconocible como una *derivada* (con respecto a la variable independiente x), lo que resta son dos integraciones sencillas, que dan $\mathrm{l}\ y = x^2 + C$. Por este motivo, la función $\rho(y) = 1/y$ se denomina factor integrante para la ecuación original (1).

Un **factor integrante** para una ecuación diferencial es una función $\rho(x, y)$ tal que la multiplicación de cada miembro de la ecuación diferencial por $\rho(x, y)$ produce una ecuación en la que cada miembro se reconoce como una derivada.

Con la ayuda del factor integrante adecuado, existe una técnica estándar para resolver la ecuación lineal de primer orden

$$\frac{dy}{dx} + P(x)y = Q(x) \qquad (3)$$

en el intervalo ñeque las funciones coeficientes $P(x)$ y $Q(x)$ sean continuas. Se multiplica cada miembro de la ecuación (3) por el factor integrante

$$\rho(x) = e^{\int P(x)\,dx} \qquad (4)$$

El resultado es

$$e^{\int P(x)\,dx}\frac{dy}{dx} + P(x)e^{\int P(x)\,dx}y = Q(x)e^{\int P(x)\,dx} \qquad (5)$$

ya que

$$D_x\left[\int P(x)dx\right] = P(x)$$

el lado izquierdo es la derivada del producto $y(x).e^{\int P(x)dx}$, de esta manera la ecuación (5) es equivalente a

$$D_x\left[y(x)\cdot\ e^{\int P(x)dx}\right] = Q(x)e^{\int P(x)dx}$$

La integración de ambos miembros de esta ecuación da

$$y(x)e^{\int P(x)dx} = \int (Q(x)e^{\int P(x)dx})dx + C$$

Por último, despejando y, se obtiene la solución general de la ecuación lineal de primer orden (3)

$$y(x) = e^{-\int P(x)dx}\left[\int\left(Q(x)e^{\int P(x)dx}\right)dx + C\right] \qquad (6)$$

Para resolver una ecuación de la forma de la ecuación (3) con las funciones coeficientes $P(x)$ y $Q(x)$ mostradas, se debe seguir la siguiente secuencia:

1. En primer lugar se calcula el factor integrante $\rho(x) = e^{\int P(x)dx}$.

2. Luego se multiplica ambos miembros de la ecuación diferencial por $\rho(x)$.

3. Posteriormente, se identifica el miembro izquierdo de la ecuación resultante como la derivada de un producto:

$$D_x[\rho(x)y(x)] = \rho(x)Q(x)$$

4. Finalmente, se integra esta ecuación,

$$\rho(x)y(x) = \int \rho(x)Q(x)dx + C$$

Y se despeja la y para obtener la solución general de la ecuación diferencial original.

2.3.6. MÉTODO DE SUSTITUCIÓN Y ECUACIONES EXACTAS.-

2.3.6.1. MÉTODO DE SUSTITUCIÓN.-

Para resolver una ecuación diferencial, en primer lugar se debe reconocer el tipo de ecuación (separable, por ejemplo), y a continuación se aplica un procedimiento formado por etapas específicas del tipo de ecuación que conduce a una función diferenciable, la cual satisface la ecuación. Con frecuencia, el primer paso es transformarla en otra ecuación diferencial mediante **sustitución**. Por ejemplo, en el caso que se quiera transformar la ecuación de primer orden $dy/dc = f(x, y)$ con la sustitución $y = g(x, u)$, en que u se considera función de la variable x. Si g tiene primeras derivadas parciales, entonces, la regla de la cadena da,

$$\frac{dy}{dx} = g_x(x,u) + g_u(x,u)\frac{du}{dx}$$

Al sustituir dy/dx con $f(x, y)$ y y con $g(x, u)$ en la derivada anterior, se obtiene la nueva ecuación diferencial de primer orden

$$f\big(x, g(x,u)\big) = g_x(x,u) + g_u(x,u)\frac{du}{dx}$$

que, después de despejar du/dk, tiene la forma $du/dx = F(x, u)$. Si se puede determinar una solución $u = \phi(x)$ de esta segunda ecuación, una solución de la ecuación diferencial ordinaria es $y = g(x, \phi(x))$.

Este método puede ser utilizado para resolver algunas clases de ecuaciones, como se explica a continuación.

Cuando una función tiene la propiedad

$$f(tx,ty) = t^{\alpha} f(x,y)$$

para un número real α, se dice que es una ecuación **homogénea** de grado α; por ejemplo, $f(x, y) = x^3 + y^3$ es homogénea de grado 3, porque

$$f(tx,ty) = (tx)^3 + (ty)^3 = t^3(x^3 + y') = t^3 f(x,y)$$

mientras que $f(x, y) = x^3 + y^3 + 1$ no es homogénea.

Una ecuación diferencial de primer orden,

$$M(x,y)dx + N(x,y)dy = 0 \qquad (1)$$

es homogénea si los coeficientes M y N, a la vez, son funciones homogéneas del mismo grado. En otras palabras, la ecuación (1) es homogénea si

$$M(tx,ty) = t^{\alpha} M(x,y) \quad \text{y} \quad N(tx,ty) = t^{\alpha} N(x,y)$$

Una ecuación diferencial homogénea como $M(x, y)\ dx + N(x, y)\ \ dy = 0$ se puede resolver por sustitución algebraica. Específicamente, alguna de las dos sustituciones $y= ux$, o $x = vy$, donde u y v son nuevas variables dependientes, reducen la ecuación a una ecuación diferencial separable, de primer orden. Para demostrarlo, se sustituye $y = ux$ y su diferencial, $dy = u\ dx + x$, en la ecuación (1):

$$M(x, ux)dx + N(x, ux)[u\ dx + x\ du] = 0$$

Se aplica la propiedad de homogeneidad para poder escribir

$$x^{\alpha}M(1, u)dx + x^{\alpha}N(1, u)[u\ dx + x\ du] = 0$$

o bien

$$[M(1, u) + uN(1, u)]dx + xN(1, u)du = 0$$

que da

$$\frac{dx}{x} + \frac{N(1,\ u)du}{M(1, u) + uN(1, U)} = 0$$

Se insiste en que esta fórmula no se debe memorizar; más bien, cada vez se debe aplicar el método. La demostración de que la sustitución $x = vy$ en la ecuación (1) también conduce a una ecuación separable es análoga.

La ecuación diferencial

$$\frac{dy}{dx} + P(x)Y = f(x)y^{n} \qquad (2)$$

en que n es cualquier número real, es la ecuación de **Bernoulli**. Obsérvese que cuando $n = 0$ y $n = 1$, la ecuación (2) es lineal. Cuando

$n \neq 0$ y $n \neq 1$, la sustitución $u = y^{1-n}$ reduce cualquier ecuación de la forma (2) a una ecuación lineal.

Sea la ecuación

$$x\frac{dy}{dx} + y = x^2y^2$$

Primero, se reformula la ecuación como se muestra

$$\frac{dy}{dx} + \frac{1}{x}y = xy^2$$

dividiéndola entre x. A continuación se sustituye, con $n = 2$,

$$y = u^{-1} \quad \text{y} \quad \frac{dy}{dx} = -u^{-2}\frac{du}{dx} \quad \leftarrow \quad \textit{regla de la cadena}$$

en la ecuación dada, y se simplifica. El resultado es

$$\frac{du}{dx} - \frac{1}{x}u = -x$$

El factor integrante para esta ecuación lineal en, por ejemplo $(0, \infty)$, es

$$e^{-\int dx/x} = e^{-\ln x^{-1}} = x^{-1}$$

Se integra $\frac{d}{dx}[x^{-1}u] = -1$

Y se obtiene $x^{-1}u = -x + c$, es decir, $u = -x^2 + cx$.

Como $y = u^{-1}$, entonces $y = 1/u$ y, en consecuencia, una solución de la ecuación es

$$y = \frac{1}{-x^2 + cx}$$

En este ejemplo hay que notar que no se ha llegado a la solución general de la ecuación diferencial no lineal original, porque $y = 0$ es una solución singular de esa ecuación. El asunto es seleccionar una sustitución tal que la ecuación transformada sea una que se pueda resolver. Aun cuando esto sea posible, no siempre es fácil, puede requerir una buena cantidad de ingenio o de prueba y error.

2.3.6.2. ECUACIONES EXACTAS.-

Sea $z = f(x, y)$ una función con derivadas parciales de primer orden continuas en una región del plano xy. Se denomina diferencial total de z a:

$$dz = \frac{\partial f}{\partial x}dx + \frac{\partial f}{\partial y}dy$$

Como por ejemplo:

$$z = xy \Rightarrow dz = ydx + x\,dy$$

$$z = x + sen(x + y) \Rightarrow dz = (1 + cos(x + y))dx + cos(x + y)dy$$

Una expresión diferencial

$$M(x, y)dx + N(x, y)dy$$

Es una **diferencial exacta** en una región del plano xy si corresponde a la diferencial total de alguna función f(x, y).

Ejemplos de algunas diferenciales exactas son:

$ydx + x\,dy$

$(1 + \cos(x + y))dx + \cos(x + y)dy$

Una ecuación diferencial $M(x, y) + N(x, y) = 0$ se dice que es una **ecuación exacta** si la expresión del primer miembro es una diferencial exacta.

El siguiente teorema proporciona un criterio para determinar si una diferencial es exacta.

Teorema.- Sean M(x, y) y N(x, y) continuas y con derivadas parciales de primer orden en una región del plano xy. Entonces, una condición necesaria suficiente para que

$$M(x, y)dx + N(x, y)dy$$

sea una diferencial exacta es que:

$$\frac{\partial M}{\partial y} = \frac{\partial N}{\partial x}$$

2.4. INTRODUCCIÓN AL SISTEMA DE ECUACIONES DIFERENCIALES.-

Como ya se ha mencionado en temas anteriores la modelización de determinados fenómenos reales dan lugar a una ecuación diferencial de orden uno o de orden superior. Igualmente, determinados problemas conducen a un sistema de ecuaciones diferenciales; por ejemplo, sistemas de muelles acoplados, problemas de mezclas con depósitos conectados o circuitos eléctricos compuestos por varias mallas.

Resolver un sistema de ecuaciones diferenciales resulta sencillo sólo en el caso en que las ecuaciones sean lineales y con coeficientes constantes. Toda ecuación lineal de orden mayor que uno puede transformarse en un sistema de ecuaciones lineales de orden uno.

Los métodos para resolver sistemas de ecuaciones lineales son análogos a los ya revisados para las ecuaciones diferenciales lineales. En el caso de sistemas de ecuaciones no lineales se debe recurrir a otro enfoque, como el estudio cualitativo del comportamiento de las soluciones o la resolución numérica.

Un sistema de ecuaciones diferenciales es un conjunto de una o más ecuaciones en las que aparecen una o más funciones incógnita, pero todas ellas dependiendo de una sola variable independiente.

Un sistema de ecuaciones diferenciales es una expresión de la forma

$$\begin{cases} F_1(x, y_1, y'_1, y_2, y'_2, \dots, y_m, y'_m) = 0 \\ F_2(x, y_1, y'_1, y_2, y'_2, \dots, y_m, y'_m) = 0 \\ \vdots \\ F_m(x, y_1, y'_1, y_2, y'_2, \dots, y_m, y'_m) = 0 \end{cases}$$

donde $y_1, y_2, \dots, y_m$ son funciones reales a determinar que dependen de x y F_i: $A \subseteq \mathrm{R}^{1+2m} \rightarrow \mathrm{R}$, $1 \le i \le m$, son funciones reales de varias variables. Se suele suponer que hay igual número de ecuaciones que de incógnitas de manera que todas las ecuaciones son independientes, es decir, ninguna puede deducirse de las demás. Lo que interesa son aquellos sistemas de ecuaciones diferenciales en los que se puede despejar la primera derivada de cada una de las funciones incógnita.

2.4.1. SISTEMAS DE PRIMER ORDEN Y APLICACIONES.-

El orden de un sistema de ecuaciones diferenciales es el orden de la derivada de mayor orden que aparece en el sistema.

Un sistema de ecuaciones diferenciales de primer orden es un conjunto de ecuaciones de la forma:

$$\begin{cases} \dfrac{dx_1}{dt} = f_1(t, x_1, \dots, x_n) \\ \dfrac{dx_2}{dt} = f_2(t, x_1, \dots, x_n) \\ \vdots \\ \dfrac{dx_n}{dt} = f_n(t, x_1, \dots, x_n) \end{cases} \tag{1}$$

Este tipo de ecuaciones se presenta con frecuencia en aplicaciones biológicas y físicas describiendo, en muchos casos, sistemas muy complicados ya que la rapidez de cambio de la variable x_i depende, no sólo de t y de x_i, sino también de los valores de las otras variables.

Si cada una de las funciones f_1, f_2,……, f_n en (1) es una función lineal en las variables dependientes x_1, x_2, ..., x_n, entonces se dice que el sistema de ecuaciones es lineal.

En este tema se va a tratar la teoría de los sistemas lineales y los métodos para resolverlos, que son extensiones naturales de la teoría correspondiente a las ecuaciones lineales de orden *n*.

Un sistema de ecuaciones diferenciales lineales de primer orden se dice que está expresado en forma normal si se escribe del siguiente modo:

$$x' = f(t, x, y)$$
$$y' = g(t, x, y)$$

donde f y g son funciones de las tres variables x, y y t (variable independiente del sistema). Una solución del sistema en el intervalo (a, b) es un par de funciones $x(t)$, $y(t)$ que satisfacen

$$x'(t) = f(t, x(t), y(t))$$
$$y'(t) = g(t, x(t), y(t))$$

idénticamente para todo $t \in (a, b)$.

EJEMPLO 1.- Probar que las funciones $x(t) = e^t$, $y(t) = e^{-t}$ son soluciones del sistema

$$x' = x^2y$$
$$y' = -xy^2$$

en toda la recta real.

En efecto, por una parte $x'(t) = e^t$ y $x(t)^2y(t) = e^{2t}e^{-t} = e^t$. Así pues, $x'(t) = x(t)^2y(t)$ y se satisface la primera ecuación igualmente. Además, $y'(t) = -e^{-t}$ y $-x(t)y(t)^2 = -e^te^{-2t} = -e^{-t}$. Por lo que, $y'(t) = -x(t)y(t)^2$ con lo que se satisface la segunda ecuación. En consecuencia, las funciones $x(t) = e^t$, $y(t) = e^{-t}$ son soluciones del sistema.

Los sistemas no tienen por qué ser de dos ecuaciones. En general, un sistema de n ecuaciones diferenciales con n funciones incógnitas $x_1(t)$,

$x_2(t), \ldots\ldots\ldots, x_n(t)$ en la variables independiente t, tendrá la siguiente forma:

$$\begin{aligned} x'_1 &= f_1(t, x_1, x_2, \ldots, x_n) \\ x'_2 &= f_2(t, x_1, x_2, \ldots, x_n) \\ &\vdots \\ x'_n &= f_n(t, x_1, x_2, \ldots, x_n) \end{aligned}$$

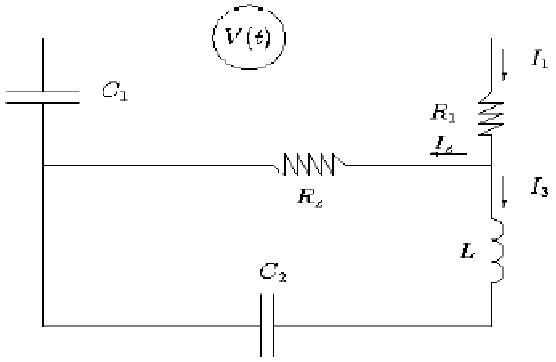
donde $f_1, f_2, \ldots, f_n$ son funciones de las $n+1$ variables $t, x_1, x_2, \ldots, x_n$. Es una condición exigible que el número de ecuaciones y de incógnitas sea el mismo. Y a este número común se lo denomina la **dimensión** del sistema.

2.4.1.1. APLICACIONES.-

Existen muchos problemas típicos que nos llevan a sistemas de ecuaciones diferenciales de primer orden como circuitos eléctricos compuestos por varios recorridos, problemas de mezclas en tanques conectados y problemas de calentamiento o enfriamiento de edificios.

CIRCUITOS ELÉCTRICOS CON VARIAS RAMAS.-

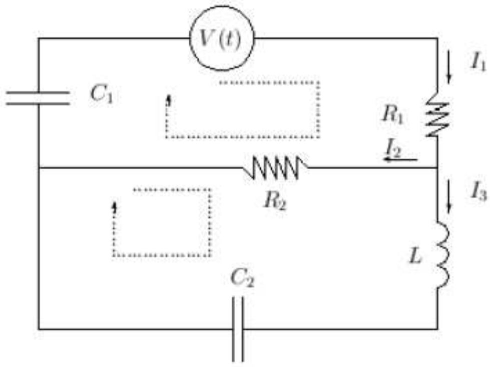
Sistemas de ecuaciones diferenciales lineales también aparecen cuando se consideran circuitos eléctricos con varias ramas o recorridos, como se muestra en la siguiente figura



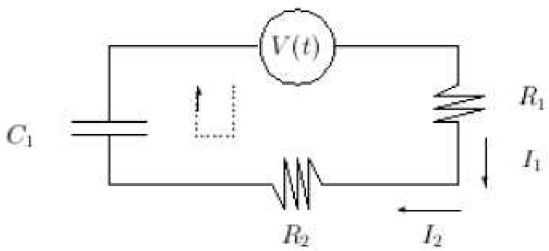
EJEMPLO 1.- En este caso se debe aplicar las leyes de Kirchoff para obtener las ecuaciones. La primera de ellas afirma que en cada nudo o punto de ramificación del circuito, la suma de las intensidades entrantes es igual a la suma de las intensidades salientes. En el circuito de la figura esto nos proporciona la ecuación

$$I_1 = I_2 + I_3.$$

En segundo lugar, se consideran los dos subcircuitos que hay y se fija un sentido de la corriente, como muestra la siguiente figura



Se toma el primer subcircuito por separado, que es



Para este subcircuito se tiene la ecuación

$$V(t) = V_{C_1} + V_{R_1} + V_{R_2}$$

donde

$$V_{C_1} = \frac{q_1}{C_1}$$

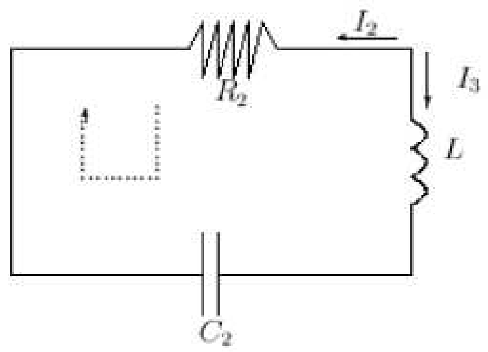
donde q_1 es la carga que da lugar a la intensidad I_1,

$$V_{R_1} = I_1 R_1$$
$$V_{R_2} = I_2 R_2$$

Teniendo en cuenta que las intensidades I_1 e I_2 llevan el sentido que se ha prefijado, y tomando la derivada primera, se obtiene la ecuación

$$V'(t) = I'_1 R_1 + {}^{I_1}/_{C_1} + I'_2 R_2$$

Se toma ahora el segundo subcircuito que muestra la figura



cuya ecuación será

$$0 = -V_{R_2} + V_L + V_{C_2},$$

teniendo en cuenta que ahora I_2 va en sentido contrario al prefijado al inicio, de ahí el signo negativo. Procediendo como antes se obtiene la ecuación

$$0 = -I_2' R_2 + I_3'' L + I_3/C_2,$$

y combinando las tres ecuaciones se tiene el sistema

$$\begin{cases} I_1 = I_2 + I_3, \\ V'(t) = I_1' R_1 + I_1/C_1 + I_2' R_2, \\ 0 = -I_2' R_2 + I_3'' L + I_3/C_2. \end{cases}$$

Eliminando I_1 se obtienen las dos ecuaciones

$$\begin{cases} V'(t) = I'_2(R_1 + R_2) + I'_3 R_1 + {}^{I_2}/_{C_1} + {}^{I_3}/_{C_1} \\ 0 = -I'_2 R_2 + I''_3 L + {}^{I_3}/_{C_2} \end{cases}$$

e introduciendo la variable $z = I'_3$, el sistema queda

$$\begin{cases} V'(t) = I_2'(R_1 + R_2) + zR_1 + I_2/C_1 + I_3/C_1, \\ I_3' = z \\ 0 = -I_2' R_2 + z' L + I_3/C_2. \end{cases}$$

Despejando I'_2, I'_3 y z' y tenemos el sistema en la forma

$$\begin{cases} I'_2 = -\dfrac{1}{C_1(R_1 + R_2)} I_2 - \dfrac{1}{C_1(R_1 + R_2)} I_3 - \dfrac{1}{R_1 + R_2} z + \dfrac{V'(t)}{R_1 + R_2} \\ I'_3 = z \\ z' = -\dfrac{R_2}{C_1 L(R_1 + R_2)} I_2 - \left(\dfrac{R_2}{C_1 L(R_1 + R_2)} + \dfrac{1}{LC_2}\right) I_3 - \dfrac{R_2 R_1}{L(R_1 + R_2)} z \end{cases}$$

que en forma matricial es

$$\begin{pmatrix} I'_2 \\ I'_3 \\ z' \end{pmatrix} = A \cdot \begin{pmatrix} I_2 \\ I_3 \\ z \end{pmatrix} + \begin{pmatrix} \dfrac{V'(t)}{R_1 + R_2} \\ 0 \\ \dfrac{R_2 V'(t)}{L(R_1 + R_2)} \end{pmatrix}$$

donde

$$A=\begin{pmatrix} -\dfrac{1}{C_1(R_1+R_2)} & -\dfrac{1}{C_1(R_1+R_2)} & -\dfrac{R_1}{R_1+R_2} \\ 0 & 0 & 1 \\ -\dfrac{R_2}{C_1L(R_1+R_2)} & -\dfrac{R_2}{C_1L(R_1+R_2)}-\dfrac{1}{LC_2} & -\dfrac{R_2R_1}{L(R_1+R_2)} \end{pmatrix}$$

2.4.2. MÉTODO DE ELIMINACIÓN.-

El método de eliminación sistemática para resolver sistemas de ecuaciones diferenciales con coeficientes constantes se basa en el principio algebraico de eliminación de variables. Se verá que la operación análoga de multiplicar una ecuación algebraica por una constante es operar en una ecuación diferencial ordinaria con cierta combinación de derivadas.

Como regla orientativa, el método de eliminación resulta eficaz en los siguientes casos:

1. En sistemas de dos ecuaciones de primer orden (sistemas 2x2), homogéneos o no, sin necesidad de que sean cero algunos coeficientes.
2. En sistemas, homogéneos o no, con ecuaciones de diversos órdenes que tengan muchos coeficientes iguales a cero, como el sistema $x'' = y$, $y'' = x$.
3. Un sistema 3x3 ó 4x4 que no tenga coeficientes iguales a cero será, en general, de resolución larga y laboriosa por cualquier método. Existe una amplia gama de casos. Si el sistema tiene autovalores complejos y/o múltiples y/o es no homogéneo, es posible que el método de eliminación resulte de interés, ya que reduce el tratamiento de la casuística (por ejemplo, de autovalores múltiples) a una ecuación.

La eliminación de una incógnita en un sistema de ecuaciones diferenciales lineales se agiliza al escribir una vez más cada ecuación del sistema en notación de un operador diferencial. Una sola ecuación de la forma

$$a_n y^{(n)} + a_{n-1} y^{(n-1)} + \cdots + a_1 y' + a_0 y = g(t)$$

Donde las a_i, i = 0, 1, ….., n son constantes, puede escribirse de nuevo como

$$\left(a_n D^{(n)} + a_{n-1} D^{(n-1)} + \cdots + a_1 D + a_0\right) y = g(t)$$

Si el operador diferencial de n –ésimo orden menor, entonces los factores conmutan.

EJEMPLO 1.-

Escribir otra vez el siguiente sistema en términos del operador *D*

$$x'' + 2x' + y'' = x + 3y + sent$$
$$x' + y' = 4x + 2y + e^{-t}$$

Solución:

Primero, se reúnen en un lado los términos con variables dependientes, luego se agrupan las mismas variables y posteriormente se reescribe cada una de las ecuaciones del sistema en notación del operador diferencial (*D*).

$$x'' + 2x' - x + y'' - 3y = sent$$
$$x' - 4x + y' - 2y = e^{-t}$$

$$(D^2 + 2D - 1)x + (D^2 - 3)y = sent$$
$$(D - 4)x + (D - 2)y = e^{-t}$$

Notación de operador diferencial

2.4.2.1. PROCEDIMIENTO PARA EL MÉTODO DE SOLUCIÓN DE SISTEMAS DE ECUACIONES DIFERENCIALES ORDINARIAS LINEALES POR ELIMINACIÓN SISTEMÁTICA.-

EJEMPLO 2.-

Por el método de eliminación sistemática resolver el siguiente sistema de ecuaciones lineales de primer orden.

$$\frac{dx}{dt} = 3y$$
$$\frac{dy}{dt} = 2x$$

Paso 1: Se cambia el sistema de la notación Leibniz a la notación Prima

$$x' = 3y$$
$$y' = 2x$$

Paso 2: Se reúnen en un lado los términos con variables dependientes y se agrupan las mismas variables.

$$x' - 3y = 0$$
$$2x - y' = 0$$

Paso 3: Se reescribe el sistema en términos del operador *D*. (Sistema transformado).

$$Dx - 3y = 0 \quad (I)$$
$$2x - Dy = 0 \quad (II)$$

Paso 4: Se elimina la variable *x*, multiplicando la ecuación (I) por el coeficiente de *x* de la ecuación (II) y, multiplicando la ecuación (II) por el coeficiente de *x* de la ecuación (I), tratando de que al multiplicarse ambas ecuaciones queden con coeficientes de signo contrario para poder eliminarlos realizando la suma.

$$(-2)Dx - 3y = 0 \qquad -2\cancel{Dx} + 6y = 0$$
$$(D)\,2x - Dy = 0 \qquad \cancel{2Dx} - D^2y = 0$$
$$-D^2y + 6y = 0$$

$$D^2y - 6y = 0 \longrightarrow (D^2 - 6)y = 0 \longleftarrow \boxed{\text{Ecuación factorizada}}$$

Paso 5: La ecuación factorizada obtenida en el paso anterior se resuelve utilizando la ecuación característica (método de las *m*) para obtener las raíces.

$$m^2 - 6 = 0 \rightarrow m^2 = 6 \rightarrow m_1 = D_1 = \sqrt{6}, m_2 = D_2 = -\sqrt{6}$$

Paso 6: Observando el tipo de raíces obtenidas en el paso anterior se decide la solución complementaria *y*(*t*) que se tendrá. (En este caso la solución complementaria corresponde al de raíces reales y distintas).

$$y(t) = c_1 e^{m_1 t} + c_2 e^{m_2 t}$$
$$y(t) = c_1 e^{\sqrt{6}t} + c_2 e^{-\sqrt{6}t} \qquad (A)$$

Paso 7: Al realizar nuevamente los pasos **4**, **5** y **6**, se elimina la variable *y* y se obtiene *x*(*t*).

$$x(t) = c_3 e^{\sqrt{6t}} + c_4 e^{-\sqrt{6t}} \qquad (B)$$

Paso 8: Sustituyendo las ecuaciones obtenidas en los pasos **6** (A) y **7** (B) en la ecuación (I) del sistema transformado obtenido en el paso **3** se calculan C_3 y C_4 en términos de C_1 y C_2, respectivamente.

$$\sqrt{6}c_3 e^{\sqrt{6t}} - \sqrt{6}c_4 e^{-\sqrt{6t}} - 3(c_1 e^{\sqrt{6t}} - c_2 e^{-\sqrt{6t}} = 0$$
$$\left(\sqrt{6}c_3 - 3c_1\right)e^{\sqrt{6t}} + \left(-\sqrt{6}c_4 + 3c_2\right)e^{-\sqrt{6t}} = 0$$
$$\left(\sqrt{6}c_3 - 3c_1\right) = 0 \qquad \left(-\sqrt{6}c_4 + 3c_2\right) = 0$$

$$\sqrt{6}c_3 = 3c_1 \rightarrow c_3 = \frac{3}{\sqrt{6}}c_1 \rightarrow c_3 = \frac{3}{\sqrt{6}}\frac{\sqrt{6}}{\sqrt{6}}c_1 = \frac{3\sqrt{6}}{6}c_1 = \frac{\sqrt{6}}{2}c_1 \rightarrow \boxed{c_3 = \frac{\sqrt{6}}{2}c_1}$$
$$-\sqrt{6}c_4 = 3c_2 \rightarrow c_4 = \frac{3}{-\sqrt{6}}c_2 \rightarrow c_4 = \frac{3}{-\sqrt{6}}\frac{-\sqrt{6}}{-\sqrt{6}}c_2 = \frac{-3\sqrt{6}}{6}c_2 = -\frac{\sqrt{6}}{2}c_2 \rightarrow \boxed{c_4 = \frac{\sqrt{6}}{2}c_2}$$

Paso 9: Se encuentra la solución del sistema original en términos de C_1 y C_2.

$$x(t) = \frac{\sqrt{6}}{2}c_1 e^{\sqrt{6t}} - \frac{\sqrt{6}}{2}c_2 e^{-\sqrt{6t}}, \;\; y(t) = c_1 e^{\sqrt{6t}} + c_2 e^{-\sqrt{6t}}$$

2.4.3. MÉTODOS NUMÉRICOS PARA SISTEMAS.-

Los métodos numéricos para el estudio del comportamiento de sistemas dinámicos han cobrado fuerza por varias razones. Probablemente la más importante, es que puede accederse a computadoras altamente eficientes a un costo cada vez más bajo, lo que permite su uso para la resolución de problemas altamente complejos. Una segunda razón es que los métodos numéricos son en muchos casos la única alternativa posible para la resolución de los frecuentes problemas no-lineales muchas veces intratables analíticamente. Por otra parte los problemas lineales continúan creciendo en magnitud, requiriendo un mayor esfuerzo para su solución.

Los métodos numéricos para la resolución de sistemas de ecuaciones diferenciales de primer orden son simplemente generalizaciones de los métodos para una sola ecuación de primer orden. Hay dos tipos de métodos para la resolución de sistemas de ecuaciones diferenciales lineales:

- Directos, proporcionan la solución exacta (salvo errores de redondeo) en un número finito de pasos: Gauss.
- Iterativos, proporcionan una sucesión $\{x_k\}$ que aproxima, o converge, a la solución exacta $\{x_k\} \to x$. El cálculo se detiene cuando se alcanza un cierto nivel de precisión.

2.4.3.1. MÉTODOS DIRECTOS.-

a) Método de Cramer.-

Es aplicable si el sistema tiene igual número de ecuaciones que de incógnitas $n = m$ y el determinante de la matriz de coeficientes es distinto de cero. Es decir, un sistema de Cramer es, por definición, compatible determinado *y*, por tanto, tiene siempre solución única.

El valor de cada incógnita x_i se obtiene de un cociente cuyo denominador es el determinante de la matriz de coeficientes, y cuyo numerador es el determinante que se obtiene al cambiar la columna *i* del determinante anterior por la columna de los términos independientes.

EJEMPLO 1.- Resolver el siguiente sistema compatible determinado:

$$\begin{array}{rcrcrcr} x & + & y & + & z & = & 11 \\ 2x & - & y & + & z & = & 5 \\ 3x & + & 2y & + & z & = & 24 \end{array}$$

La matriz aumentada del sistema sería

$$\left(\begin{array}{ccc|c} 1 & 1 & 1 & 11 \\ 2 & -1 & 1 & 5 \\ 3 & 2 & 1 & 24 \end{array}\right)$$

Se comprueba que el determinante del sistema es diferente de cero:

$$|A| = \begin{vmatrix} 1 & 1 & 1 \\ 2 & -1 & 1 \\ 3 & 2 & 1 \end{vmatrix} = (-1 + 3 + 4) - (3 + 2 + 2) = 5$$

$$|A_x| = \begin{vmatrix} 11 & 1 & 1 \\ 5 & -1 & 1 \\ 24 & 2 & 1 \end{vmatrix} = (-11 + 24 + 10) - (-24 + 22 + 5) = 20$$

$$|A_y| = \begin{vmatrix} 1 & 11 & 1 \\ 2 & 5 & 1 \\ 3 & 24 & 1 \end{vmatrix} = (5 + 33 + 48) - (15 + 24 + 22) = 25$$

$$|A_z| = \begin{vmatrix} 1 & 1 & 11 \\ 2 & -1 & 5 \\ 3 & 2 & 24 \end{vmatrix} = (-24 + 15 + 44) - (-33 + 10 + 48) = 10$$

$$x = \frac{|A_x|}{|A|} = 4 \qquad y = \frac{|A_y|}{|A|} = 5 \qquad z = \frac{|A_z|}{|A|} = 2$$

b) Método de matriz inversa.-

Es aplicable si el sistema tiene igual número de ecuaciones que de incógnitas $n = m$ y el determinante de la matriz de coeficientes es distinto de cero. Es decir, resuelve sistemas compatibles determinados (no homogéneos).

$$A * x = b$$

$$A^{-1} * A * x = A^{-1} * B$$

$$1 * x = A^{-1} * B$$

$$x = A^{-1} * B$$

EJEMPLO 2.- Solucionar el sistema propuesto en el ejemplo anterior mediante el método de matriz inversa.

$$\begin{array}{rcrcrcr} x & + & y & + & z & = & 11 \\ 2x & - & y & + & z & = & 5 \\ 3x & + & 2y & + & z & = & 24 \end{array}$$

$$A^{-1} = \frac{1}{|A|} \; (Adf(A)^t)$$

$$|A| = \begin{vmatrix} 1 & 1 & 1 \\ 2 & -1 & 1 \\ 3 & 2 & 1 \end{vmatrix} = 5 \; ; \; (Adf(A)^t) = \begin{pmatrix} -3 & 1 & 2 \\ 1 & -2 & 1 \\ 7 & 1 & -3 \end{pmatrix}$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \frac{1}{5} \begin{pmatrix} -3 & 1 & 2 \\ 1 & -2 & 1 \\ 7 & 1 & -3 \end{pmatrix} \begin{pmatrix} 11 \\ 5 \\ 24 \end{pmatrix} = \begin{pmatrix} 4 \\ 5 \\ 2 \end{pmatrix}$$

c) Método De Gauss.-

El método de Gauss, también conocido como método de eliminación simple de Gauss, es una de las primeras técnicas empleadas por actuarios, matemáticos e ingenieros para la resolución de sistemas de ecuaciones. El método comprende dos fases:

- Eliminación de las incógnitas hacia delante.
- Sustitución hacia atrás.

La primera fase tiene el objetivo de reducir el sistema original a una forma triangular superior. Por ejemplo, para un sistema de n ecuaciones con n incógnitas que se representa con la siguiente matriz aumentada:

$$\left(\begin{array}{cccc|c} a_{11}x_1 & a_{12}x_2 & \cdots & a_{1n}x_n & b_1 \\ a_{21}x_1 & a_{22}x_2 & \cdots & a_{2n}x_n & b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1}x_1 & a_{n2}x_2 & \cdots & a_{nn}x_n & b_{\mathrm{n}} \end{array}\right)$$

Una vez que se tiene la matriz aumentada, se realizan una serie de transformaciones para convertir los elementos bajo la diagonal superior en ceros:

$$\left(\begin{array}{cccc|c} a_{11}x_1 & a_{12}x_2 & \cdots & a_{1n}x_n & b_1 \\ 0 & a_{22}x_2 & \cdots & a_{2n}x_n & b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{nn}x_n & b_{\mathrm{n}} \end{array}\right)$$

La segunda fase de la eliminación de Gauss, consiste en que, una vez que se ha obtenido una matriz triangular superior a través de operaciones de normalización, se realiza la sustitución hacia atrás. Este proceso comienza despejando x_n de la última ecuación (último renglón de la matriz).

$$x_n = \frac{b_n^{(n\text{-}1)}}{a_{nn}^{(n\text{-}1)}}$$

A su vez, este resultado se sustituye hacia atrás en la ecuación n-1 del sistema modificado final (renglón n-1 de la matriz). Este mecanismo se repite para las x restantes, lo que se representa mediante la expresión general:

$$x_i = \left(b_i^{(i\text{-}1)} - \sum_{j=i+1}^{n} a_{ij}^{(i\text{-}1)} x_j \right) \Big/ a_{ii}^{(i\text{-}1)}$$

para $i = n$-1, n-2, ….., 1.

EJEMPLO 3.- Resolver el siguiente sistema de ecuaciones mediante el método de Gauss.

$$\begin{array}{rcrcrcr} & & 2x_2 & + & 3x_3 & = & 0 \\ 4x_1 & + & 6x_2 & + & 7x_3 & = & -3 \\ 2x_1 & + & x_2 & + & 6x_3 & = & 5 \end{array}$$

A continuación, se construye la matriz aumentada; pero antes, se reordenan las ecuaciones de tal forma que no haya ceros en la diagonal principal:

$$\left(\begin{array}{ccc|c} 4 & 6 & 7 & -3 \\ 0 & 2 & 3 & 0 \\ 2 & 1 & 6 & 5 \end{array}\right) \xrightarrow{F_3=F_3-F_1/2} \left(\begin{array}{ccc|c} 4 & 6 & 7 & -3 \\ 0 & 2 & 3 & 0 \\ 0 & -2 & 5/2 & 13/2 \end{array}\right) \xrightarrow{F_3=F_3-F_2} \left(\begin{array}{ccc|c} 4 & 6 & 7 & -3 \\ 0 & 2 & 3 & 0 \\ 0 & 0 & 11/2 & 13/2 \end{array}\right)$$

$$x_3 = 13/11$$

$$x_2 = -3\ (13/17)/2 = -39/22$$

$$x_1 = (-\ 3 + 6\ (39/22) - 7\ (13/11))\ /\ 4 = -\ 7/44$$

$$x = \begin{pmatrix} -\ 7/44 \\ -39/22 \\ 13/11 \end{pmatrix}$$

d) Método de Gauss-Jordan.-

El método de Gauss-Jordan, también denominado eliminación de Gauss-Jordan, es un método por el cual pueden resolverse sistemas de ecuaciones lineales con n número de variables, encontrar matrices y matrices inversas.

Para resolver sistemas de ecuaciones lineales aplicando este método, en primer lugar se debe anotar los coeficientes de las variables del sistema de ecuaciones lineales en su notación matricial:

$$\begin{array}{ccccccc} a_1x & + & b_1y & + & c_1z & = & d_1 \\ a_2x & + & b_2y & + & c_2z & = & d_2 \\ a_3x & + & b_3y & + & c_3z & = & d_3 \end{array}$$

Entonces, anotando como matriz (también denominada matriz aumentada):

$$\left(\begin{array}{ccc|c} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \end{array}\right)$$

A continuación, se procede a convertir dicha matriz en una matriz identidad, es decir, una matriz equivalente a la original, la cual es de la forma:

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

Esto se logra aplicando a las distintas filas y columnas de las matrices simples operaciones de suma, resta, multiplicación y división; teniendo en cuenta que una operación se aplicará a todos los elementos de la fila o de la columna, cualquiera que sea el caso.

En dicha matriz identidad no aparecen los términos independientes, esto se debe a que cuando la matriz original alcance la forma de la matriz identidad, dichos términos resultarán ser la solución del sistema y verificarán la igualdad para cada una de las variables, correspondiéndose de la siguiente forma:

- $d_1 = x$
- $d_2 = y$
- $d_3 = z$

EJEMPLO 4.- Sea el sistema:

$$\begin{array}{rcrcrcr} 2x_1 & + & 3x_2 & + & 4x_3 & = & 6 \\ 4x_1 & + & 5x_2 & + & 10x_3 & = & 16 \\ 4x_1 & + & 8x_2 & + & 2x_3 & = & 2 \end{array}$$

$$\left(\begin{array}{ccc|c} 2 & 3 & 4 & 6 \\ 4 & 5 & 10 & 16 \\ 4 & 8 & 2 & 2 \end{array}\right) \xrightarrow{F_1=F_1/2} \left(\begin{array}{ccc|c} 1 & 3/2 & 2 & 3 \\ 4 & 5 & 10 & 16 \\ 4 & 8 & 2 & 2 \end{array}\right) \xrightarrow[F_3=F_3-4F_1]{F_2=F_2-4F_1} \left(\begin{array}{ccc|c} 1 & 3/2 & 2 & 3 \\ 0 & -1 & 2 & 4 \\ 0 & 2 & -6 & -10 \end{array}\right)$$

$$\xrightarrow{F_2=-F_2} \left(\begin{array}{ccc|c} 1 & 3/2 & 2 & 3 \\ 0 & 1 & -2 & -4 \\ 0 & 2 & -6 & -10 \end{array}\right) \xrightarrow[F_3=F_3-2F_2]{F_1=F_1-3F_2/2} \left(\begin{array}{ccc|c} 1 & 0 & 5 & 9 \\ 0 & 1 & -2 & -4 \\ 0 & 0 & -2 & -2 \end{array}\right)$$

$$\xrightarrow{F_3=-F_3/2} \left(\begin{array}{ccc|c} 1 & 0 & 5 & 9 \\ 0 & 1 & -2 & -4 \\ 0 & 0 & 1 & 1 \end{array}\right) \xrightarrow[F_3=F_3-2F_2]{F_1=F_1-3F_2/2} \left(\begin{array}{ccc|c} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 1 \end{array}\right)$$

$$x = \begin{pmatrix} 4 \\ -2 \\ 1 \end{pmatrix}$$

e) Factorización LU.-

Suponer que la matriz A es una matriz n x n, ésta se puede escribir como el producto de dos matrices:

$$A = LU$$

Donde L es una matriz triangular inferior n x n y U es una matriz escalonada n x n. Entonces para resolver el sistema:

$$Ax = b$$

que equivale a:

$$Ax = (LU)x = L(Ux)$$

Una posible estrategia de solución consiste en tomar $y = Ux$ y resolver para y:

$$Ly = b \qquad (1)$$

Como la matriz L es triangular inferior este sistema puede resolverse mediante sustitución hacia abajo, una vez encontrados los valores de y, el sistema inicial se resuelve despejando x de:

$$Ux = y \qquad (2)$$

Nuevamente, como U es escalonada, este sistema puede resolverse (en caso de tener solución) mediante sustitución hacia atrás.

EJEMPLO 5.- Resolver el sistema propuesto en el ejemplo anterior mediante la factorización LU:

$$\begin{array}{rcrcrcr} 2x_1 & + & 3x_2 & + & 4x_3 & = & 6 \\ 4x_1 & + & 5x_2 & + & 10x_3 & = & 16 \\ 4x_1 & + & 8x_2 & + & 2x_3 & = & 2 \end{array}$$

Cuya matriz de coeficiente es:

$$A = \begin{pmatrix} 2 & 3 & 4 \\ 4 & 5 & 10 \\ 4 & 8 & 2 \end{pmatrix} = LU$$

El vector de solución es:

$$b = \begin{pmatrix} 6 \\ 16 \\ 2 \end{pmatrix}$$

Para hallar la matriz U se convierte la triangular inferior de la matriz A en ceros; para ello se utiliza Gauss, simultáneamente se calcula L tomando la matriz identidad y ubicando el opuesto del número por el cual se multiplica en la transformación de U debajo de la diagonal principal de L, como se muestra a continuación:

$$\xrightarrow[F_3=F_3-2F_1]{F_2=F_2-2F_1} \begin{pmatrix} 2 & 3 & 4 \\ 0 & -1 & 2 \\ 0 & 2 & -6 \end{pmatrix} \xrightarrow{F_2=F_2+2\,F_2} \begin{pmatrix} 2 & 3 & 4 \\ 0 & -1 & 2 \\ 0 & 0 & -2 \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & 0 & 0 \\ * & 1 & 0 \\ * & * & 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 2 & * & 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 2 & -2 & 1 \end{pmatrix}$$

Una vez hecha la factorización de la matriz se procede a hallar la solución del sistema empleando la ecuación (1):

$$\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 2 & -2 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 16 \\ 2 \end{pmatrix}$$

$$y_1 = 6$$

$$y_2 = 16 - 2y_1 = 4$$

$$y_3 = 2 + 2y_2 - 2y_1 = -2$$

Por lo tanto, se tiene:

$$y = \begin{pmatrix} 6 \\ 4 \\ -2 \end{pmatrix}$$

Ahora, se emplea la ecuación (2):

$$\begin{pmatrix} 2 & 3 & 4 \\ 0 & -1 & 2 \\ 0 & 0 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 4 \\ -2 \end{pmatrix}$$

$$x_3 = 1$$

$$x_2 = \frac{4 - 2x_3}{1} = -2$$

$$x_2 = \frac{6 - 3x_2 - 4x_3}{2} = 4$$

Por lo que la solución del sistema lineal dado es:

$$x = \begin{pmatrix} 4 \\ -2 \\ 1 \end{pmatrix}$$

2.4.3.2. MÉTODOS ITERATIVOS.-

a) Método de Jacobi.-

El método Jacobi es el método iterativo para resolver sistemas de ecuaciones lineales más simples y se aplica sólo a sistemas cuadrados, es decir, a sistemas con tantas incógnitas como ecuaciones.

- Primero se determina la ecuación de recurrencia. Para ello se ordenan las ecuaciones y las incógnitas. De la ecuación i se despeja la incógnita i. En notación matricial se escribe como:

$$x = c + Bx \qquad (1)$$

  donde x es el vector de incógnitas.

- Se toma una aproximación para las soluciones y a ésta se le designa x0.

- Se itera en el ciclo que cambia la aproximación:

$$x_{i+1} = c + Bxi \qquad (2)$$

EJEMPLO 7.- Partiendo de ($x = 1$; $y = 2$) aplicar 2 iteraciones del método de Jacobi para resolver el sistema:

$$\begin{aligned} 5x + 2y &= 5 \\ x - 4y &= 0 \end{aligned}$$

Solución: Primero, se debe despejar de la ecuación la incógnita correspondiente:

$$x = 0.20 + 0.00x - 0.40y$$

$$y = 0.00 + 0.25x + 0.00y$$

Se aplica la primera iteración partiendo de $x_0 = 1.00$ y $y_0 = 2.00$:

$$x_1 = 0.20 + 0.00\ (1.00) - 0.40\ (2.00) = -0.60$$

$$y_1 = 0.00 + 0.25\ (1.00) + 0.00\ (2.00) = 0.25$$

Se emplea la segunda iteración, partiendo de $x_1 = -0.60$ y $y_1 = 0.25$:

$$x_2 = 0.20 + 0.00\ (-0.60) - 0.40\ (0.25) = 0.10$$

$$y_2 = 0.00 + 0.25\ (-0.60) + 0.00\ (0.25) = -0.15$$

Se prosigue con la siguiente iteración, partiendo de $x_2 = 0.10$ y $y_2 = -0.15$:

$$x_3 = 0.20 + 0.00\ (0.10) - 0.40\ (-0.15) = 0.26$$

$$y_3 = 0.00 + 0.25\ (0.10) + 0.00\ (-0.15) = 0.025$$

Se continúa con la siguiente iteración partiendo de $x_3 = 0.26$ y $y_3 = 0.025$:

$$x_4 = 0.20 + 0.00\ (0.26) - 0.40\ (0.025) = 0.190$$

$$y_4 = 0.00 + 0.25\ (0.26) + 0.00\ (0.025) = 0.065$$

Se asigna la siguiente iteración partiendo de $x_4 = 0.190$ y $y_4 = 0.025$:

$$x_5 = 0.20 + 0.00\ (0.190) - 0.40\ (0.065) = 0.174$$

$$y_5 = 0.00 + 0.25\ (0.190) + 0.00\ (0.065) = 0.0475$$

Se aplica la siguiente iteración partiendo de $x_5 = 0.174$ y $y_5 = 0.025$:

$$x_6 = 0.20 + 0.00\ (0.174) - 0.40\ (0.025) = 0.181$$

$$y_6 = 0.00 + 0.25\ (0.174) + 0.00\ (0.025) = 0.0435$$

$$\boldsymbol{x = 0.181}$$

$$\boldsymbol{y = 0.0435}$$

b) Método Gauss-Seidel.-

Este es uno de los más interesantes métodos del análisis numérico y particularmente útil, ya que permite **encontrar la solución de un sistema de "*n*" ecuaciones con "*n*" incógnitas**.

Como se trata de un método iterativo, debe aplicarse recursivamente hasta hallar una solución adecuada o con un error considerablemente pequeño.

En cada iteración se obtiene una posible solución del sistema con un error determinado, a medida que se aplica nuevamente el método, la solución puede ser más precisa, entonces se dice que el sistema converge, pero si al aplicar el método reiteradas veces la solución tiene un error cada vez mayor, se dice que el sistema de ecuaciones no converge y no se puede resolver con este método.

Sea el siguiente sistema de ecuaciones:

$$
\begin{array}{ccccccccc}
a_{11}x_1 & + & a_{12}x_2 & + & ---- & + & a_{1z}x_n & = & b_1 \\
a_{21}x_1 & + & a_{22}x_2 & + & ---- & + & a_{2z}x_n & = & b_2 \\
\vdots & & \vdots & & \vdots & & \vdots & & \vdots \\
a_{x1}x_1 & + & a_{x2}x_2 & + & ---- & + & a_{xz}x_n & = & b_n
\end{array}
$$

Se despeja x_1 de la ecuación 1, x_2 de la ecuación 2, x_n de la ecuación n, entonces queda:

$$x_1 = \frac{b_1 - a_{22}x_2 - \ldots\ldots.. - a_{1z}x_n}{a_{11}}$$

$$x_2 = \frac{b_2 - a_{21}x_1 - \ldots\ldots.. - a_{2z}x_n}{a_{22}}$$

$$x_n = \frac{b_\mathrm{n} - a_{11}x_1 - \ldots\ldots.. - a_{x\text{-}1}x_{n\text{-}1}}{a_\mathrm{xz}}$$

De la fórmula anterior provienen las fórmulas a aplicarse en las diferentes iteraciones. Para empezar aplicar el método se debe asignar un valor arbitrario a las variables x_2, ……., x_n con el fin de obtener x_1. Lo más conveniente en este caso, es que los valores empiecen en cero, lo cual

$$x_1 = \frac{b_1}{a_{11}}$$

facilita el trabajo pues se reduce el cálculo de las primeras soluciones, entonces:

Ahora, se despeja x_2 de la ecuación 2 y se reemplaza a x_1 por el valor obtenido de la ecuación anterior. De lo que se obtiene:

$$x_2 = \frac{b_2 - a_{21}\left(\dfrac{b_1}{a_{11}}\right)}{a_{22}}$$

Una vez obtenido x_2, se despeja x_3 de la ecuación 3 y así sucesivamente con las n ecuaciones que se tengan, cada vez asignando el valor de las x_1, x_2, …., x_{n-1} obtenido en el paso anterior.

Cuando ya se ha despejado las x_n, se tiene lo que se denomina primera solución o solución de la primera iteración.

$$\begin{aligned} x_1 &= \alpha_1 \\ x_2 &= \alpha_2 \\ &\vdots \\ x_n &= \alpha_n \end{aligned}$$

Con los nuevos valores de x_1, x_2, …., x_n se aplican los mismos pasos anteriores pero con los valores actuales de las x_n, de esta manera se consigue una segunda solución:

$$\begin{aligned} x_1 &= \beta_1 \\ x_2 &= \beta_2 \\ &\vdots \end{aligned}$$

$$x_n = \beta_n$$

Al contar ya con esta segunda solución, se tienen condiciones para poder calcular el error y se realiza de la siguiente manera:

$$|\in_{n,1}| = \frac{\beta_1 - \alpha_1}{\beta_1} \text{ x } 100\%$$

$$|\in_{n,2}| = \frac{\beta_2 - \alpha_2}{\beta_2} \text{ x } 100\%$$

$$|\in_{n,n}| = \frac{\beta_n - \alpha_n}{\beta_n} \text{ x } 100\%$$

De esta forma se repite el método, tantas veces, hasta que el error sea mínimo o lo suficientemente aceptable.

Ahora, para que un sistema sea convergente se debe cumplir que la matriz de coeficientes sea diagonalmente dominante, y para ello se debe verificar la expresión que sigue:

$$|\alpha_{11}| > \sum_{j=1} |\alpha_{ij}|$$

$$i = 1, 2, \ldots., n$$

Si no se cumple esa condición, se puede permutar las filas de la matriz, con el fin de poder convertirla en una diagonalmente dominante.

EJEMPLO 8.- Emplear el método de Gauss-Seidel para aproximar la solución del sistema:

$$5x + 2y = 5$$

$$x - 4y = 0$$

Solución: En primer lugar se despeja de la ecuación la incógnita correspondiente:

$$x = 0.20 + 0.00x - 0.40y$$

$$y = 0.00 + 0.25x + 0.00y$$

Se designa la primera iteración partiendo de $x_0 = 1.00$ y $y_0 = 2.00$:

$$x_1 = 0.20 + 0.00\,(1.00) - 0.40\,(2.00) = \mathbf{-0.60}$$

$$y_1 = 0.00 + 0.25\,(\mathbf{-0.60}) + 0.00\,(2.00) = -0.15$$

Se prosigue con la segunda iteración partiendo de $x_1 = -0.60$ y $y_1 = -0.15$:

$$x_2 = 0.20 + 0.00\,(-0.60) - 0.40\,(-0.15) = \mathbf{0.26}$$

$$y_2 = 0.00 + 0.25\,(\mathbf{0.26}) + 0.00\,(-0.15) = 0.065$$

Se aplica la tercera iteración partiendo de $x_2 = 0.26$ y $y_2 = 0.065$:

$$x_3 = 0.20 + 0.00\,(0.26) - 0.40\,(0.065) = \mathbf{0.174}$$

$$y_3 = 0.00 + 0.25\,(\mathbf{0.174}) + 0.00\,(0.065) = 0.0435$$

$$\boldsymbol{x = 0.174}$$

$$\boldsymbol{y = 0.0435}$$

Como se puede observar, con el método de Gauss-Seidel se redujeron el número de iteraciones en comparación al método de Jacobi.

CAPÍTULO 3

3. INTRODUCCIÓN A MATLAB

MATLAB (*Matrix Laboratory*) es un lenguaje de alto nivel, diseñado para proveer facilidades de cálculos numéricos, visualización y programación en un entorno muy sencillo de utilizar. MATLAB ofrece un entorno de desarrollo integrado (IDE) muy versátil con un lenguaje de programación propio (lenguaje M). Proporciona una interfase con líneas de comando para resolver problemas lineales y no lineales, además de otros experimentos en forma numérica. Utiliza una notación matemática simple para tratar los problemas y resolverlos. Su poder radica en el manejo de matrices en forma eficiente, también incluye su propio compilador lo cual permite extender su uso permitiendo al usuario crear sus propios comandos, clases y funciones. Es compatible con uno de los más usados lenguajes de programación: C y Fortran.

Se trata de un software matemático para cálculo científico (aritmético y simbólico) basado en matrices. Entre sus prestaciones básicas se hallan:

- Realizar cálculos aritméticos (como una calculadora).
- Realizar cálculo simbólico (con la posibilidad de hacer operaciones como derivar funciones, calcular primitivas, etc.) y exacto.
- Programar en un lenguaje no compilado (script).
- Realizar gráficos en dos y tres dimensiones.
- La representación de datos y funciones.
- La implementación de algoritmos.
- La creación de interfaces de usuario (GUI).
- La comunicación con programas en otros lenguajes y con otros dispositivos hardware.

Las capacidades de MATLAB se pueden ampliar con las llamadas "cajas de herramientas" o *toolboxes*. Así, por ejemplo, se pueden utilizar *toolboxes* de Matemáticas y Optimización, Estadística y Análisis de datos, Diseño de sistemas de control y análisis, Procesado de señal y comunicaciones, Procesado de imagen, Pruebas y medidas, Biología computacional, Redes Neuronales Artificiales, Modelado y análisis financiero, Desarrollo de aplicaciones o informes y conexión a bases de datos, lo que facilita el estudio de sistemas dinámicos y su regulación.

3.1. GENERALIDADES.-

MATLAB es un intérprete de comandos que permite efectuar cualquier operación con matrices de forma sencilla y rápida. Es un sistema interactivo cuyo elemento básico es la matriz (y como caso particular, vectores o escalares), tanto de datos numéricos como de información no numérica. En el MATLAB todo elemento es considerado como una matriz: por ejemplo, un escalar es una matriz de (1x1); un vector es una matriz con solo una fila o una columna; una cadena de caracteres es una matriz fila de elementos (uno por letra), etc.

Al ser un intérprete, en la ventana principal aparece evaluado el último comando suministrado al programa. Para evitar su visualización, conviene terminar los comandos con punto y coma ";". Esto es también útil cuando se ejecutan programas con varias órdenes, si no se desea que todos los resultados de las operaciones sean volcados en la pantalla.

Si no se especifica el nombre de una variable para almacenar un resultado, por defecto este se vuelca en una variable temporal "*ans*". Todas las variables que se utilicen durante una sesión quedan en el espacio de trabajo de MATLAB, y pueden utilizarse mientras no se salga del programa o se eliminen. Para revisar las variables existentes en el espacio de trabajo de MATLAB, se utiliza *who* o *whos*. Al visualizar la salida de esta orden, se puede destacar el uso de las variables como matrices y la distinción de si tienen o no elementos complejos.

En la mayoría de los sistemas, la forma de invocar a MATLAB es con el comando del shell '*matlab*'. MATLAB muestra un mensaje inicial indicando que está listo para aceptar instrucciones y haciendo algunas sugerencias de comandos para iniciar. A partir de allí se pueden escribir comandos inmediatamente.

Si se presenta algún tipo de dificultad, se puede interrumpir la tarea que está realizando MATLAB con ‘Control – C’ (usualmente escrito como ‘C – c’ para abreviar). Para salir de Matlab simplemente se debe escribir ‘*exit*’ en el *prompt*. En aquellos sistemas que soportan control de tareas (como Linux y la mayoría de los sistemas Unix, VMS, etc.) se puede suspender MATLAB (u Octave según sea el caso) enviando una señal ‘SIGSTP’ (usualmente ‘C–z’).

Si se novato en el uso de MATLAB, es recomendable tratar de reproducir los ejemplos que se muestran. Las líneas marcadas como ‘>>‘ son líneas que se deben escribir terminándolas con un retorno de carro (tecla ‘enter’ de la PC). MATLAB responderá con un resultado o un gráfico.

Una vez iniciado el programa, la primera ventana que se muestra contiene pequeñas ventanas en su interior, entre ellas las denominadas Ventana de Comando (*Command Window*), Ventana del Directorio Actual (*Current Directory*) y la Ventana del Histórico de Comandos (*Command History*). En la figura siguiente se muestra la visión por defecto que se presenta en MATLAB.

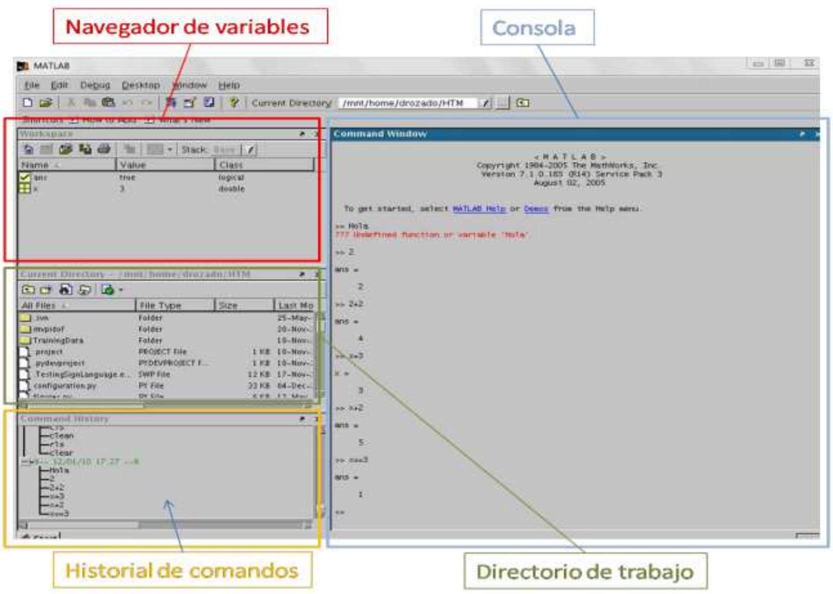


Figura 6.- Entorno MATLAB
Fuente: Autor.

3.2. CARACTERÍSTICAS DE LA VENTANA DE COMANDOS.-

La Ventana de Comandos es la ventana principal de MATLAB, se abre cuando se inicia la aplicación. Las funciones introducidas (o "entradas") se ejecutan pulsando la tecla *Enter*. Al escribir los nombres de las funciones o de los comandos, es importante recordar que MATLAB distingue entre mayúsculas y minúsculas (habitualmente, las funciones se escriben en minúsculas).

Asimismo, tras seleccionar una zona de esta ventana, el botón derecho del ratón despliega un menú emergente que permite, entre otras opciones, evaluar dicha selección, e igualmente, abrirla con el *Editor/Debugger* como un M-fichero.

Seleccionar *File → Preferences*... permite especificar el formato numérico a emplear y otras opciones de presentación en pantalla. De igual forma, es posible seleccionar el tipo y el color de las fuentes de texto.

3.2.1. PRINCIPALES COMANDOS DE MATLAB.-

Los principales comandos de la ventana de comandos son los que a continuación se describen:

Comandos	Descripción
» load	Lee todas o algunas de las variables de un fichero.
» open	Abre, entre otros, los ficheros .mat, M-ficheros o ficheros .fig de gráficos. Un fichero también puede abrirse seleccionando *File → Open*... De modo equivalente, las variables pueden importarse eligiendo *File → Import Data*...

» clear	Elimina algunas o todas las variables del espacio de trabajo. Igualmente, *Edit → Clear Workspace* elimina todas las variables del espacio de trabajo.
» clc	Borra la *ventana de comandos* (no elimina las variables). Este comando equivale a seleccionar *Edit → Clear Workspace*.
» format modo	Determina el formato de salida en la *ventana de comandos*. Entre los distintos "modos", pueden destacarse: short (muestra hasta 5 dígitos) long (muestra hasta 15 dígitos) y rat (formato racional).
» cd	Permite conocer y cambiar el directorio actual.
» cd..	Disminuye un nivel en el árbol de carpetas. El directorio puede ser igualmente modificado en la *ventana de directorio actual.*
» who	Muestra un listado con las variables del espacio de trabajo. Estas variables aparecen, igualmente, en la *ventana de espacio de trabajo*.
» dir	Muestra un listado con los archivos del directorio actual. Esta información también es asequible a través de la *ventana de directorio actual*.
» edit M-fichero	Abre una *ventana de edición* con un M-fichero. Si no se especifica un M-fichero la *ventana de edición* se abre en blanco. Igualmente puede seleccionarse *File → New → M-file*, o hacer clic en el botón  de la barra de herramientas.
» save	Guarda todas o algunas de las variables del espacio de trabajo. Análogamente, puede seleccionarse *File → Save Workspace As*...
» exit/quit	Cierra el programa MATLAB. Igualmente es posible cerrar el programa mediante *File → Exit MATLAB*.

Tabla 1.- Principales comandos de MATLAB.

Algunas teclas o combinaciones de teclas resultan especialmente interesantes en la ventana de comandos:

- Las teclas ↑ y ↓ permiten recuperar comandos escritos con anterioridad.
- La tecla Esc elimina todo el texto escrito en una línea.
- La combinación de teclas Control + c aborta la ejecución de una sentencia.

3.3. MANEJO DEL WORKSPACE.-

Cuando se trabaja en la *ventana de* órdenes, MATLAB recuerda tanto los nombres y valores de las variables como las órdenes introducidas. Estas órdenes y variables forman el llamado **espacio de trabajo** de MATLAB y pueden ser recuperadas cuando se requiera. Entonces, el *Workspace* no es más que el espacio de memoria que utiliza Matlab.

Todas las variables definidas se almacenan y se muestran en el *Workspace*, además de sus características, como el tipo de variable y su tamaño.

Esta ventana permite, a través de la herramienta *Array editor*, editar y modificar las matrices que constituyen las diversas variables. Asimismo, cada una de estas variables puede representarse gráficamente, de forma sencilla, haciendo clic con el botón derecho y seleccionando *Graph Selection* ►

Las preferencias relativas a esta ventana pueden modificarse en *File* → *Preferences*...

Además de crear y borrar variables, resulta útil guardar sesiones de trabajo, es decir, todas las instrucciones que se digitan en la *ventana de instrucciones*. Para ello, al inicio de la sesión se ingresa *diary <nombre del fichero>*. También se puede elegir el tipo de archivo, aunque es recomendable que éste sea de texto (por ejemplo, *diary c:\regulación\resumen.txt*). Para detener la grabación de la sesión se digita *diary off*.

Del mismo modo, se puede almacenar únicamente el valor de algunas variables:

» *save c:\regulación\datos x y*

No es necesario añadir la extensión, MATLAB automáticamente creará el fichero con extensión .mat. Las variables x e y han de estar previamente definidas.

Para recuperar las variables guardadas se ingresa:

» *load datos*

Si no se recuerda el nombre de alguna variable o se desea conocer cuántas variables hay y qué nombres tienen, se puede solicitar a MATLAB que liste las variables *who* y *whos* (información más completa que *who*).

Para recordar órdenes previas se puede utilizar las teclas del cursor para recuperar las órdenes previas y mover el cursor dentro de ellas. De esta manera se consigue volver a llamar órdenes para una nueva evaluación y editarlas para su corrección.

3.4. FORMATOS DE NÚMEROS.-

Por defecto, un número entero se visualiza como entero; un número real aparece con cuatro decimales salvo si los dígitos significativos se encuentran fuera de este rango, en cuyo caso se visualiza con notación científica. Se puede cambiar el formato de salida seleccionando el menú *Preferences/General/Numerical Format* en el menú *File*. Otra posibilidad es utilizar la orden apropiada. El cambio de formato no cambia la representación interna, sólo la visualización.

Cabe destacar que, con el formato que aparece por defecto, algunos números pueden aparecer por pantalla como 0.0000, aunque realmente no son 0 (sino $3.5*10^{-7}$, por ejemplo).

Orden de Matlab	**Comentarios**	**Ejemplo**
format short	Visualización por defecto	3.5833e+01
format long	16 dígitos	3.583333333333334
format short e	5 dígitos más exponente	3.5833e+01
format long e	16 dígitos más exponente	3.583333333333334e+01
format hex	Hexadecimal	4041eaaaaaaaaaab
format bank	2 decimales	35.83
format +	Signo +	+
format rat	Aproximación racional	215/6

Tabla 2.- Formatos de números.

Las capacidades simbólicas de la *toolbox de matemática simbólica* permiten realizar operaciones con un número arbitrario de dígitos, sin embargo, a mayor número de dígitos, mayor tiempo y memoria requerirán las operaciones.

La orden *digits(n)* cambia el número de dígitos de precisión que se usa por defecto en la *toolbox*. La orden *digits* permite conocer cual es el valor de este número. Por último, la orden *vpa* permite realizar un cálculo y mostrar su resultado con una precisión especificada sin cambiar el número de dígitos de precisión con el que se trabaja por defecto.

```
>>format long
>>pi
ans =
        3.14159265358979
>>digits
Digits = 32
```

```
>>vpa('pi')       %evalúa pi con 32 dígitos
ans =
        3.1415926535897932384626433832795
>>vpa('pi',60)
ans =
        3.14159265358979323846264338327950288419716939937510582097494
>>digits(50)
>>pi
ans =
        3.14159265358979
>>vpa('pi')
ans =
        3.1415926535897932384626433832795028841971693993751
>>digits
Digits = 50
```

3.5. GESTIÓN DE DIRECTORIOS.-

Por defecto, al arrancar MATLAB se empieza en el directorio matlab\bin, mientras que los ficheros (*M-files* o modelos de SIMULINK) suelen estar en otras carpetas.

Para "moverse" en MATLAB, se utilizará los comandos típicos de MS-DOS: *cd* para acceder a un directorio, *dir* para ver el contenido de un directorio, *cd*.. para salir de una directorio.

Dentro del entorno de Matlab se puede cambiar de directorio, borrar y crear archivos y directorios, visualizar los contenidos de directorios, etc.

Los comandos más comunes son:

Comandos	Descripción

ls ó dir	Para listar los contenidos del directorio actual. Permite el uso de comodines (ej. *ls* *.m muestra todos los archivos con extensión *.m*, mientras que *ls* ???.m solamente muestra los archivos con extensión *.m* y que su nombre esté compuesto de tres caracteres.
cd	Para cambiar de un directorio a otro.
pwd	Para mostrar el directorio actual (esta información también se muestra en la barra).
mkdir	Para crear un directorio (ej. *mkdir directorio1* crear un directorio llamado directorio1 dentro del directorio actual).
rmdir	Para borrar un directorio, el mismo debe estar vacío. (ej. *rmdir directorio1* borra el directorio de nombre directorio1). Si a *rmdir* se le añade el parámetro *s*, se borra el contenido del directorio y todo subdirectorio que exista dentro del mismo (ej. *rmdir directorio1 s*).
delete	Se utiliza para borrar un archivo concreto (ej. *delete archivo1* borra el archivo1 del directorio actual).

Tabla 3.- Comando para gestión de directorios.

3.6. M-FILES.-

Aunque MATLAB se utiliza como un sofisticado calculador, su verdadera potencia radica en su capacidad para leer y ejecutar ficheros escritos por el usuario. Sin embargo, cuando se van a realizar una serie de operaciones más complicadas y de forma repetitiva, se utilizan los llamados ficheros *M* que tienen una extensión *.m*. Es una buena práctica escribir comandos directamente en un fichero *M* y guardarlos en un subdirectorio apropiado. Si se comete un error, es mucho más fácil editar el fichero para encontrarlo que buscar el error entre varios comandos y, además, un fichero *M* puede guardarse y recuperarse en otro momento.

Hay fundamentalmente dos tipos diferentes de ficheros *M*: los ficheros *script* y los ficheros *function*. Un fichero *script* contiene una sucesión de comandos MATLAB que se ejecutan cuando el nombre del fichero se introduce en la ventana de comandos del sistema. Un fichero *function* requiere una entrada que procesa y devuelve el resultado como salida. MATLAB contiene un gran número de ficheros *function* en su interior.

Esos ficheros van desde las funciones elementales (por ejemplo, *exp* y *sin*) hasta un gran número de funciones especiales. Todos los ficheros *function* están caracterizados por su capacidad para coger una entrada de la línea de comandos y devolver una salida en la ventana de comandos sin que haya necesidad de editar el fichero. Un ejemplo sencillo de un fichero *function* es la función exponencial ***exp***(***x***) es decir, e^x. Se introduce

y = *exp*(*1*)

y se obtiene

y = 2.7183

Si se introduce

y = *exp*(*0*)

la salida es

y = 1

3.6.1. CREACIÓN Y UTILIZACIÓN DE UN FICHERO *M*.-

3.6.1.1. CREACIÓN DE UNA CARPETA PERSONAL.-

En primer lugar, hay que crear una carpeta en la que se van a guardar los ficheros ***M***. Para ello, hay que ubicar un subdirectorio apropiado dentro de **"*Current Directory*"** en la ventana de MATLAB. Dentro de la ventana ***Current Directory*** se pulsa el botón **"*New Folder*"** y se le pone un nombre apropiado, por ejemplo "*Mis ficheros*".

3.6.1.2. CREACIÓN DE UN FICHERO *M*.-

La manera más ágil de crear un fichero ***M*** es seleccionar la opción **"*New M-File*"** situada en la parte superior izquierda de la pantalla de *MATLAB* debajo del botón **"*File*"**. De forma alternativa, presionando **"*File - New - M-File*"** se obtiene el mismo resultado. Esta acción proporciona una ventana similar a la que se muestra a continuación:

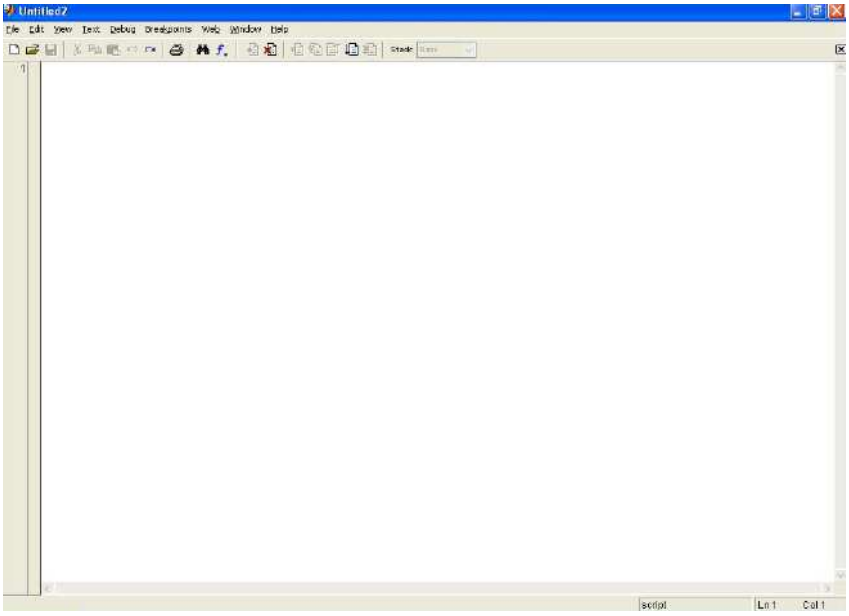


Figura 7.- Ventana activa de un nuevo fichero *M*.
Fuente: http://portalevlm.usal.es/Portal/e_books/guiaalumno/Chapter24SG_Spanish.pdf.

Para demostrar de una manera clara lo expresado anteriormente, se puede abrir un fichero ***M*** en MATLAB y guardarlo con un nombre diferente, como por ejemplo "***MAT1.m***", en la carpeta que se haya creado anteriormente. Esta es una práctica excelente para asegurarse de no estar duplicando un nombre de algún fichero ya existente. Una manera fácil de lograr este cometido, es introducir el comando

help MAT1

en la ventana de comandos. El sistema devuelve

***MAT1.m not found*.**

Si por el contrario se ingresa

help sin

el sistema devuelve

***SIN Sine*.**

***SIN*(*X*) *is the sine of the elements of X*.**

Overloaded methods

help sym/sin.m

lo que indica que ya existe un fichero ***M*** que se llama **"*sin.m*"**.

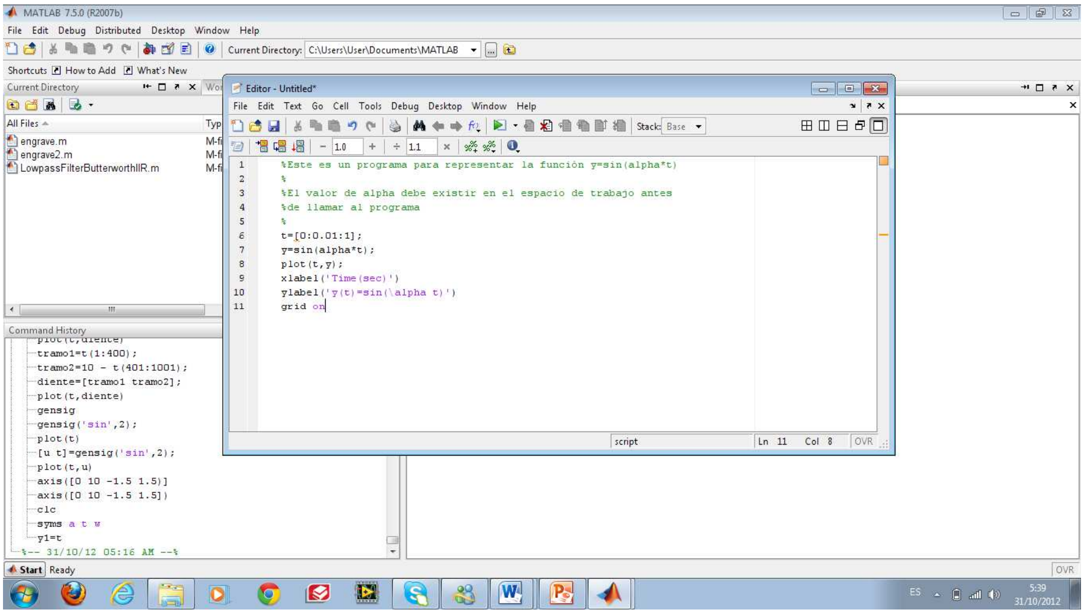


Figura 8.- Ejemplo del editor para crear un fichero *M*.
Fuente: Autor

a) CREACIÓN DE UN FICHERO *SCRIPT*.-

Para muestra, se va a crear un sencillo fichero *script* para efectuar la suma de dos números. En la línea 1 del fichero recientemente creado se digita

y=3+2

(obsérvese que no se ha colocado **";"** puesto que no se podría visualizar la salida del sistema si se inserta dicho símbolo) y se selecciona la opción **"*File - Save As*"** para guardar el fichero como ***MAT1*** en la carpeta. Ahora, al presionar **"*Debug - Run*"** aparecerá la siguiente pantalla:

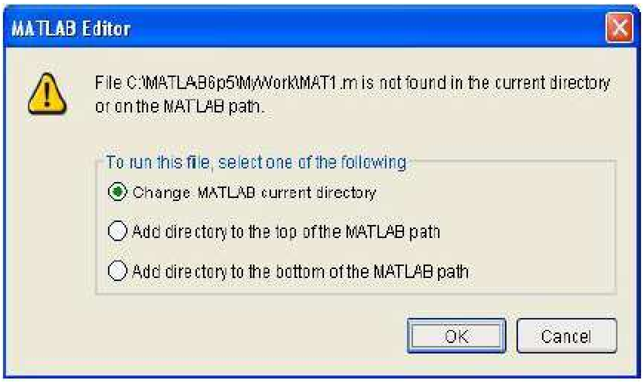


Figura 9.- Mensaje mostrado al crear un nuevo fichero ***script***.
Fuente: http://portalevlm.usal.es/Portal/e_books/guiaalumno/Chapter24SG_Spanish.pdf.

Para conseguir que el sistema reconozca la carpeta se presiona *OK*. Si ahora se introduce

MAT1

La salida es

y=5

que es lo que se esperaba. Si, por el contrario aparece el mensaje

***??? Undefined function or variable 'MAT1'*.**

quiere decir que no se ha seleccionado ***OK*** en la ventana anterior.

Si se requiere realizar un cambio en la entrada, se abre y edita el fichero. Por ejemplo, si se desea cambiar el fichero ***MAT1*** para calcular 3+3, el contenido del fichero debe cambiarse por

y=3+3

y guardar el fichero a continuación. Introduciendo ahora ***MAT1*** en la línea de comando se obtiene la respuesta esperada

y=6

Es una buena práctica introducir comentarios en los ficheros que expliquen lo que el cada uno realiza. Estos comentarios suelen introducirse al comienzo del fichero. Para ello basta con editar el fichero, colocarse en la zona superior izquierda y pulsar **[** para que la primera línea del código se desplace a la siguiente línea. Cada línea de comentario comienza necesariamente con **"%"**. Eso le dice al sistema que ignore el contenido de la línea y vaya a la primera línea que no comience por ese símbolo. Por ejemplo, se puede insertar en la primera línea del fichero el comentario:

%MAT1 – Calcula la suma de dos números

Obsérvese que, de manera automática, el sistema cambia el color de escritura y escribe en verde los comentarios. Eso facilita la depuración del programa si es necesario. Si se ejecuta el fichero, se obtiene la misma respuesta que en el caso anterior. Si, por el contrario, se ha olvidado comenzar la línea con **"%"** el sistema devuelve:

??? Error: File: add.m Line: 2 Column: 18

Unexpected MATLAB expresión

y no se lleva a cabo el cálculo.

b) CREACIÓN DE UN FICHERO *FUNCTION*.-

Ahora se procede a crear un fichero ***function*** para sumar dos números. Se crea otro fichero ***M*** y se lo guarda en la carpeta. Ahora se introduce:

```
function [out] = add(x,y)
%ADD - Calculates the sum of two numbers
out=x+y;
```

y se guarda el fichero. Se procede a escribir el comentario después de haber definido la función. Ahora se introduce

***add*(2,3)**

y la salida es

***ans*=5**

Puede ponerse un nombre a la salida, por ejemplo **"*y*"**, escribiendo

***y*=*add*(2,3)**

y la salida es

y=5

3.7. OPERACIONES CON VECTORES.-

Para definir una matriz o un vector no hace falta establecer de antemano su tamaño.

MATLAB puede operar con matrices por medio de operadores y por medio de funciones. Los operadores matriciales de MATLAB son los siguientes:

Símbolo	Descripción
+	Adición o suma
-	Sustracción o resta
*	Multiplicación
′	Traspuesta
^	Potenciación
\	División izquierda
/	División derecha
.*	Producto elemento a elemento
. /	División elemento a elemento
. ^	Elevar a una potencia elemento a elemento

Tabla 4.- Operadores matriciales de MATLAB.

Estos operadores se aplican también a las variables o valores escalares, aunque con algunas diferencias.

Todos estos operadores son coherentes con las correspondientes operaciones matriciales: por ejemplo, no se puede sumar matrices que no sean del mismo tamaño. Si los operadores no son utilizados de manera correcta, se obtiene un mensaje de error.

Los operadores se pueden aplicar también de modo mixto, es decir con un operando escalar y otro matricial. En este caso la operación con el escalar se aplica a cada uno de los elementos de la matriz. Considérese el siguiente ejemplo:

```
>> A=[1   2; 3   4]
A =
1     2
3     4
>> A*2
ans =
2     4
6     8
>> A-4
ans =
-3    -2
-1     0
```

MATLAB utiliza el operador de división / para dividir por un escalar todos los elementos de una matriz o vector.

3.7.1. OPERACIONES ENTRE VECTOR Y ESCALAR.-

Las operaciones entre escalares y vectores son directas. Desde el punto de vista teórico, no se puede sumar un escalar a un vector. Sin embargo, MATLAB sí lo permite.

Por ejemplo, si *y* es un vector, el comando *y+2* añadirá 2 a cada componente del vector. Al ingresar los siguientes comandos

```
>> y=1:5
```

>> *y+2*

>> *y-2*

>> *2*y*

>> *y/2*

Por supuesto, estas operaciones son igualmente válidas para vectores columna

>> *w*=(*1*:*3*:*20*)'

>> *w+3*

>> *w-11*

>> *.1*w*

>> *w/10*

3.7.2. OPERACIONES ENTRE VECTORES.-

Considérense los siguientes vectores

>> *a=1*:*3*

>> *b=4*:*6*

La adición y sustracción de vectores es natural y fácil. Introduciendo los siguientes comandos

>> *a*,*b*,*a+b*

>> *a*,*b*,*a-b*

Como no aparece punto y coma que suprima la salida, el comando *a*,*b*,*a+b* mostrará primero el vector *a*, luego el vector *b* y por último el *a+b*.

Estas operaciones son válidas para vectores columna

>> *a*=(*1*:*3*)’,*b*=(*4*:*6*)’
>> *a*+*b*,*a*-*b*

Sin embargo, se pueden obtener resultados no esperados si no se recuerda que MATLAB es un entorno que trabaja con matrices.

>> *a*,*b*,*a***b*

El último comando devuelve un error porque * es el símbolo de MATLAB para la multiplicación de matrices, y en este caso hay un problema de compatibilidad entre las órdenes de las “matrices” *a* y *b*. También pueden ocurrir errores si se intenta añadir vectores de diferente tamaño

>> *a*=*1*:*3*,*b*=*4*:*7*,*a*+*b*

3.7.3. OPERACIONES CON COMPONENTES.-

Para multiplicar los vectores *a* y *b* componente a componente, ejecútese el siguiente comando de MATLAB

>> *a*=(*1*:*3*)’,*b*=(*4*:*6*)’
>> *a*,*b*,*a*.**b*

El símbolo .* es el operador de MATLAB para la multiplicación elemento a elemento. La salida se calcula multiplicando las primeras componentes de los vectores *a* y *b*, a continuación las segundas componentes, etc. El operador de MATLAB para la división componente a componente es ./

>> *a*,*b*,*a*./*b*

Para elevar cada componente de un vector a una potencia, se utiliza .^

>> *a*,*a*.*^2*

3.8. REPRESENTACIONES GRÁFICAS.-

MATLAB ofrece gran número de posibilidades a la hora de realizar representaciones gráficas. La representación de cualquier serie de datos es uno de los puntos fuertes de MATLAB. Dispone de funciones para representar series de puntos, superficies, curvas de nivel, entre otros. Permite agrupar y superponer representaciones. Todo ello con variaciones de estilo y de coordenadas. Permite a su vez realizar gráficos de tipo estadístico: de barra, histogramas, etc. Prácticamente cualquier cosa puede representarse gráficamente en MATLAB.

Por las características propias del programa, los gráficos, en concreto los 2D, están orientados a la representación gráfica de vectores. Se utiliza una ventana especial para la creación de los gráficos: la ventana gráfica o de dibujo y, dichos gráficos se guardan en ficheros de extensión *.fig*. Ciertos comandos ejecutados sobre la línea de comandos son los que abren esta ventana, otros dibujan sobre la ventana activa, bien sustituyendo lo que había en ella, bien añadiendo nuevos elementos gráficos a los que había.

3.8.1. FUNCIONES BÁSICAS PARA LAS GRÁFICAS.-

Existe una orden que evalúa cuidadosamente la función que se va a representar y asegura que todas sus peculiaridades se representen en la gráfica de salida. El comando básico para la representación de gráficos 2D es el comando *plot*. Su sintaxis puede ser:

plot(*x,y*): dibuja el conjunto de puntos (*x*,*y*) donde las abscisas de los puntos se encuentran en el vector *x* y las ordenadas en el *y*.

Para representar una función *f*(*x*) es necesario conocer los valores de puntos de la forma (*x*,*f*(*x*)). Para ello puede seguirse alguno de estos caminos:

- Definir un vector *x* con el rango de variación donde se desea pintar la función. Para ello puede ser muy útil el comando *linspace*(*xmin*,*xmax*,*n*). Crear el vector y evaluando *f* en *x*. Por ejemplo:

```
>> x=linspace(0,10,100);
>> y=sin(x);
>> plot(x,y)
```

  Por defecto, MATLAB dibuja uniendo los puntos con línea continua de color azul y un grosor determinado, opciones todas que se podrán alterar como veremos.

- También es posible dibujar una función con el comando *fplot* cuya sintaxis es la siguiente: *fplot*('*f*(*x*)',[*xmin*,*xmax*]). Así, este comando admite como argumento un nombre de función o de un fichero *.m* en el que está definida la función a representar. Por ejemplo:

```
>> fplot('sin(x)',[-3*pi,3*pi,-1,1])
```

En general, si no se cierra la ventana de dibujo generada al evaluar un comando como los anteriores, si se vuelve a ejecutar uno de ellos, se dibuja sobre dicha ventana perdiéndose el primer dibujo. Si se desea representar varias funciones a la vez las opciones son:

- ***plot***(*x*,*y*,*x*,*z*) donde *x* el vector de las abscisas, común para las dos representaciones, *y* es el de las ordenadas de la primera representación y *z* las de la segunda.
- ***fplot***(‘[*f1*(*x*),*f2*(*x*),...]’,[*xmin*,*xmax*]) donde *f1*, *f2*, … son las funciones a representar en el intervalo de variación marcado por *xmin* y *xmax*.
- ***hold on***, ***hold off***. Todos los gráficos que se ordene dibujar entre los comandos *hold on* y *hold off* se representan en la misma figura. Si hay una figura abierta se dibujan en ésta.

Ejemplo:

```
>> hold on
>> x=[-3*pi:1:3*pi];
>> plot(x,sin(x))
>> plot(x,tan(x),'r')
>> hold off
```

- El comando ***subplot***. Una ventana gráfica se puede dividir en *m* particiones horizontales y *n* verticales para representar *m*x*n* figuras. Cada una de las particiones tendrá sus ejes aunque las propiedades serán comunes a todas ellas. La sintaxis es: *subplot*(*m*,*n*,*i*), donde *m* y *n* son el número de subdivisiones e *i* la subdivisión activa. Por ejemplo:

```
>> x=0:0.1:2*pi;
>> y=sin(x);z=cos(x);t=exp(-x);v=x^2;
>> subplot(2,2,1), plot(x,y)
>> subplot(2,2,2), plot(x,z)
>> subplot(2,2,3), plot(x,t)
>> subplot(2,2,4), plot(x,v)
```

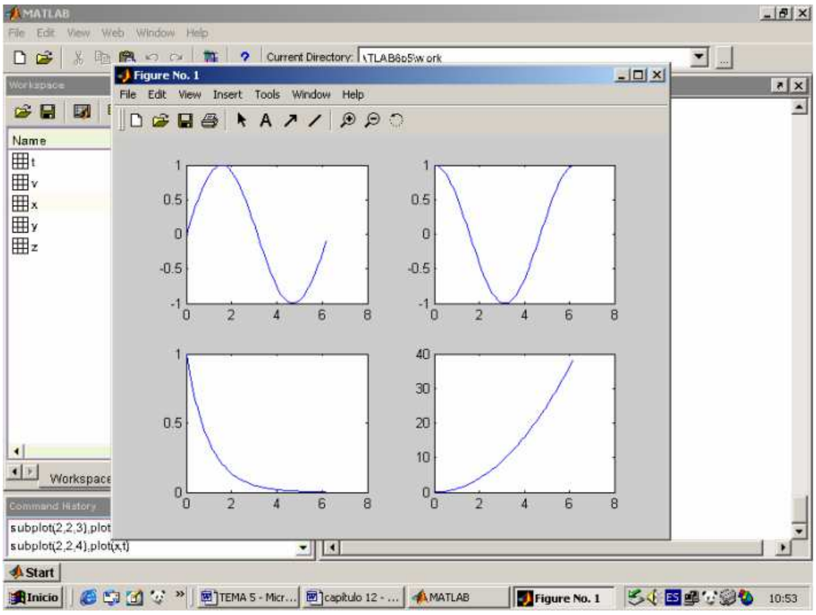


Figura 10.- Ventana gráfica de *subplot*.
Fuente:
http://www2.caminos.upm.es/departamentos/matematicas/Fdistancia/PIE/matlab/temasmatlab/TEMA%205.pdf.

3.8.2. OPCIONES DE DIBUJO.-

Como ya se ha mencionado anteriormente, el comando *plot*(*x*,*y*) dibuja los gráficos con unas características predefinidas en el programa, y es posible alterarlas a partir de *plot*(*x*,*y*,*s*) donde *s* se compone de dos dígitos entre comillas. Uno fija el color de la línea y otro el carácter a usar en el gráfico. Por ejemplo:

>> *plot*(*x*,*y*,'-*g*')

dibuja los puntos unidos con una línea continua, marcando los puntos con *, y en verde.

Otros tipos de marcadores son: . * x ó + (marcan los puntos en el gráfico).

Otros tipos de línea: los puntos se unen con una línea con las siguientes posibilidades de apariencia:

- (línea continua)
- - (línea formada por trazos discontinuos)
-. (línea formada por puntos y trazos)
: (línea formada por puntos)

Los colores vienen dados por: y: amarillo; g: verde; m: magenta; b: azul; c: cían; w: blanco; r: rojo; k: negro.

Se puede modificar el grosor de línea incluyendo la cadena: '*linewidth*', número_indicativo_del_grosor. Por ejemplo: *plot*(*x*,*y*,'*linewidth*',*2*).

En general, se puede obtener una excelente descripción del comando *plot* y otros relacionados con el mediante la ejecución de *help plot*.

3.8.3. FIGURA ACTIVA.-

Lo más importante de representar gráficos en MATLAB es el concepto de figura activa. MATLAB puede tener abiertas centenares de ventanas al mismo tiempo pero sólo podrá representar datos en una: la figura activa. Se puede controlar dicha figura mediante el comando *figure*.

El comando ***figure***(*n*) crea una nueva figura o selecciona como activa la figura dada. Cada figura tiene asignado un número entero, si *n* es una figura no existente creará una nueva y la activará, si *n* existe activará la figura correspondiente.

Otra consideración importante es que cada vez que se representan datos en la figura activa todo lo que esté ya en ella se va a borrar. Este comportamiento no es el deseado en muchos casos y puede modificarse mediante el comando *hold*.

El comando ***hold*** cambia el comportamiento de la ventana activa. Funciona como un interruptor: *hold on* hace que cada dato se represente sobre lo anterior y *hold off* borra la ventana antes de pintar en ella. Por omisión se encuentra en *off*.

Un comando útil es *clf*, que borra la figura activa.

3.8.4. TÍTULOS Y ETIQUETAS.-

MATLAB permite manejar correctamente anotaciones sobre los gráficos y los ejes mediante la colocación adecuada de títulos y etiquetas, rejillas o leyendas. El siguiente paso es poner etiquetas: un identificador para cada eje y un título si es necesario. Las etiquetas se aplicarán, sólo en la ventana activa. Los comandos más usados son:

title(*str*), añade un título a la figura activa.

xlabel(*str*), añade una etiqueta al eje *x* de la ventana activa.

ylabel(*str*), añade una etiqueta al eje *y* de la ventana activa.

Por ejemplo,

```
x = linspace(0,500,10000);
plot(x,exp(-x/100).*sin(x));
title('Una función cualquiera')
xlabel('Tiempo')
ylabel('Amplitud')
```

Con el código anterior se obtiene el siguiente gráfico:

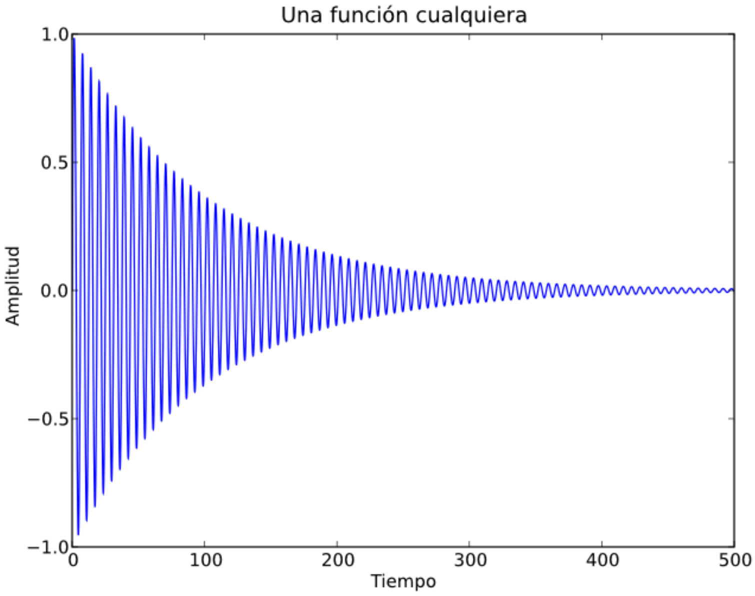


Figura 11.- Ejemplo de títulos y etiquetas.
Fuente: http://iimyo.forja.rediris.es/tutorial/.

A continuación se proporciona a los gráficos con más de una curva, de una leyenda que las distinga. Esto se consigue mediante la función *legend*.

***legend*(…)**, Al igual que con *plot* se puede utilizar esta función de múltiples maneras. La más simple es pasarle como argumento tantas cadenas de caracteres como curvas se haya representado y automáticamente asignará por orden cada curva al identificador.

Suponiendo que se quiere representar el seno hiperbólico y el coseno hiperbólico y para distinguirlos MATLAB siempre pinta la primera curva en azul y la segunda en verde. Para ello se realiza lo siguiente:

x = *linspace*(-*pi*,*pi*,*100*);

plot (*x*,*sinh*(*x*),*x*,*cosh*(*x*));

legend(‘*seno hiperbólico*’,’*coseno hiperbólico*’);

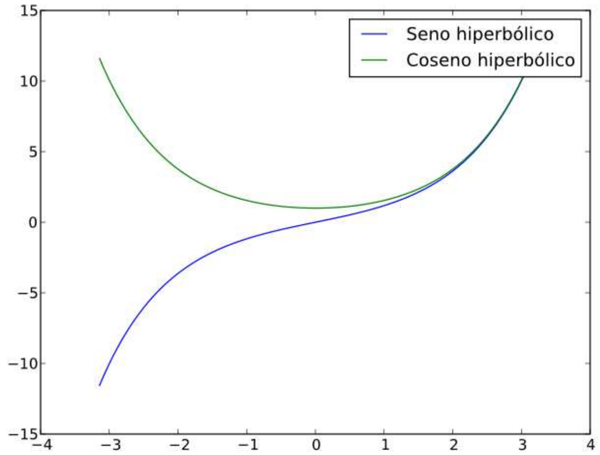


Figura 12.- Ejemplo de aplicación de la leyenda.
Fuente: http://iimyo.forja.rediris.es/tutorial/.

3.8.5. OTROS COMANDOS.-

No todas las curvas en el plano pueden representarse a partir del comando *plot* por los ejes que utiliza. En algunos casos obligatoriamente hay que utilizar otros sistemas de coordenadas o cambiar las referencias de los ejes. Las siguientes funciones pueden ser de mucha utilidad.

semilogx(*...*), representa gráficamente una serie de puntos tomando logaritmos en el eje *x*.

semilogy(*...*), representa gráficamente una serie de puntos tomando logaritmos en el eje *y*.

loglog(*...*), representa gráficamente una serie de puntos tomando logaritmos en ambos ejes.

polar(*...*), representa gráficamente una serie de datos en coordenadas polares. El primer argumento corresponde al ángulo respecto a la dirección principal θ y el segundo a la distancia respecto al centro de referencia ρ.

El uso y funcionamiento de las funciones anteriormente nombradas son idénticos a los de la función *plot*.

Un ejemplo de uso de la función polar es el siguiente:

>> *x* = *linspace*(-*pi*,*pi*,*100*);
>> *polar*(*x*,*cos*(*2*.**x*));

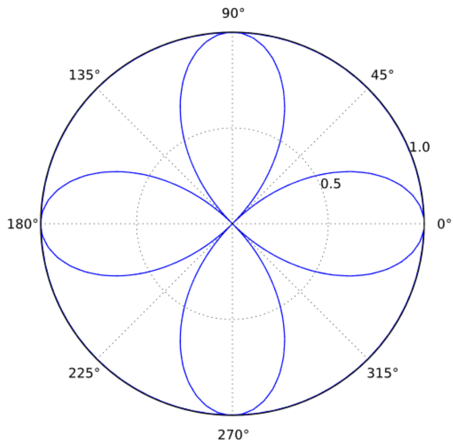


Figura 13.- Ejemplo de gráfica en coordenadas polares.
Fuente: http://iimyo.forja.rediris.es/tutorial/.

3.9. TOOLBOX DE CONTROL.-

Son funciones de aplicación específica para ingeniería de control de sistemas. Son ficheros *.M

Sirven tanto para control continuo como para control discreto, clásico (en espacios transformados sobre sistemas LTI) y de otros tipos (variables de estado, borroso, neuronal, robusto, no lineal, etc.)

En los dos campos permite realizar tareas de: **modelado**, **conversión de modelos** y **análisis** de respuesta temporal, frecuencial y en espacios transformados.

Las herramientas para obtención de los modelos de los sistemas se encuentran en otra Toolbox: la de identificación.

Todas las funciones de control se encuentran en la demo de control que se ejecuta con el comando MATLAB: *ctrldemo*.

3.10. ARITMÉTICA ELEMENTAL.-

Las operaciones habituales con números reales se introducen como en una calculadora. Para obtener un resultado se presiona la tecla ↵ . La jerarquía de las operaciones se estructura por niveles, en el siguiente orden: primero paréntesis, la operación potencia tiene el orden de prioridad más alto, seguida de productos y cocientes, y, finalmente, sumas y restas, ambas con la misma prioridad. Dentro de un mismo nivel, las operaciones se realizan en orden de izquierda a derecha.

EJEMPLO 1.- Hallar el valor de $3^2 + 23 \times 2 + 5/4 - \sqrt{14}$

Solución: Teniendo en cuenta la jerarquía, se ingresa

```
>> 3^2 + 23 *2 + 5/4 – 14^0.5
ans  =
        52.5083
```

EJEMPLO 2.- Hallar $12^{(3^4)}$

Solución:

```
>> 12^(3^4)
ans  =
        2.5923e+087
```

Si en la orden anterior se hubieran omitido los paréntesis, MATLAB realizaría la operación $(12^3)^4$.

```
>> (12^3) ^ 4   ,   12 ^ 3 ^ 4
ans  =
        8.9161e+012
ans  =
        8.9161e+012
```

Para finalizar o separar órdenes se utiliza el salto de línea, el símbolo **,** y el símbolo **;** éste último además hace que no se muestre el resultado en la pantalla (aunque internamente sí hace el cálculo). Por ejemplo:

```
>> 12 ^ 3 ^ 4;   (12^3) ^ 4;
```

El símbolo % permite introducir *comentarios* en una instrucción. MATLAB ignora todo lo que se escriba a su derecha. Por ejemplo:

```
>> 1 +2 + 3           % suma de tres números
```

3.11. ALMACENANDO RESULTADOS EN VARIABLES.-

Lo más frecuente al trabajar con MATLAB es almacenar datos, bien para recuperarlos posteriormente, bien para operar con ellos. Este almacenamiento se realiza utilizando **variables**.

Una variable es un nombre al cual se le asigna un valor. El símbolo = es el utilizado para la asignación de valores a las variables. No tiene el significado matemático de igualdad; en este caso, la última expresión sería un absurdo 0 = 4.

EJEMPLO.- Hallar $\sqrt{1245/342} + \sqrt[3]{1245/342} + \sqrt[5]{1245/342}$

Solución:

```
>> a = 1245/342
```

Con esta instrucción se almacena en *a* el resultado de la operación 1245/342. Después se puede escribir

```
>> a^0.5+a^(1/3)+a^(1/5)
```

Una variable actúa como un contenedor de datos que puede ir cambiando. Por ejemplo:

```
>> a=1;
>> a=a+4      %suma 4 al contenido de a y almacena el resultado en a
a =
      5
```

El nombre de la variable siempre debe ir a la izquierda del símbolo = y el valor a la derecha. Es muy importante tener en cuenta que el valor que se asigna a las variables es permanente. Se guardará hasta que la variable sea borrada o se le asigne un nuevo valor. Así:

```
>> x=3
x =
     3
>> x^2
x =
     9
```

Se le asigna un nuevo valor a x:

```
>> x = 7
x =
     7
>> x^2
x =
     49
```

3.12. CONSTRUCCIÓN DE MATRICES Y OPERACIONES ELEMENTALES.-

3.12.1. CONSTRUCCIÓN DE MATRICES.-

Por defecto, MATLAB trabaja con matrices de números complejos. Esto supone la ventaja substancial de no tener que declarar tipos de variable ni tamaños de fila o columnas para trabajar tanto con matrices de números reales o complejos como con vectores o escalares, que se consideran casos especiales de matrices.

Una matriz creada es guardada en una variable de manera que se pueda hacer referencia a ella más tarde. Los elementos de la matriz se introducen por filas. Los elementos de una fila se separan con comas (,) y las distintas filas por puntos y comas (;). Por ejemplo, para introducir la matriz:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ -1 & 4 & 8 \\ 7 & 2 & 1 \end{pmatrix}$$

Se escribe:

A = [1, 2, 3; -1, 4, 8; 7, 2, 1];

Para visualizar la matriz en la pantalla se ingresa

>> A $\boxed{\hookleftarrow}$.

Lo mismo que con vectores, también se puede separar los elementos de una fila con espacios y las filas pulsando la tecla [Intro].

A = [1, 2; 3, 4]

B = [–1 –2
–3 –4]

El elemento en la fila *i* y la columna *j* de la matriz *A* se denota por *A*(*i*, *j*). Se procede a modificar, por ejemplo, el elemento 2,1 de *A*:

A(2,1) = 0

A(*i* , :) denota la fila *i* de la matriz *A*. Análogamente, *A*(: , *j*) es la columna *j* de *A*.

```
A(2, :)
A(:, 1)
```

En ocasiones, resulta cómodo construir una matriz a partir de bloques. Con tal de que sus tamaños sean coherentes, basta escribir los bloques por filas, como si se tratase de elementos individuales.

```
M = [A,B;B,A]
```

Para extraer una submatriz, se indicará las filas y columnas de que se compone.

```
M41 = [1:3;2:4]
```

Las filas o columnas no tienen por qué ser consecutivas.

```
fil = [1,2,4],  col = [1,3,4]
M32 = M(fil, col)
```

MATLAB tiene varias funciones que facilitan la edición de matrices de uso frecuente:

- **eye**(n) Proporciona la matriz identidad de orden *n*.
- **zeros**(m,n) Inicializa una matriz *n* x *m* con todos los elementos nulos.
- **ones**(n,m) Realiza lo mismo con elementos de valor 1.
- **rand**(n,m) Crea matrices con elementos aleatorios uniformemente distribuidos en el intervalo [0,1].

Ejemplo:

M = eye(4)

N = zeros(3)

O = ones(2)

P = rand(3,2)

3.12.2. OPERACIONES ELEMENTALES CON MATRICES.-

Una vez definidas las matrices se pueden ejecutar algunas operaciones elementales.

Para sumar dos matrices del mismo tamaño, se suma cada elemento de una con el elemento correspondiente de la otra.

A + B

El producto de matrices se hace multiplicando fila por columna y sumando los resultados. Comprobar que el elemento (1,1) de la matriz producto AB es

A(1,1)*B(1,1)+A(1,2)*B(2,1)

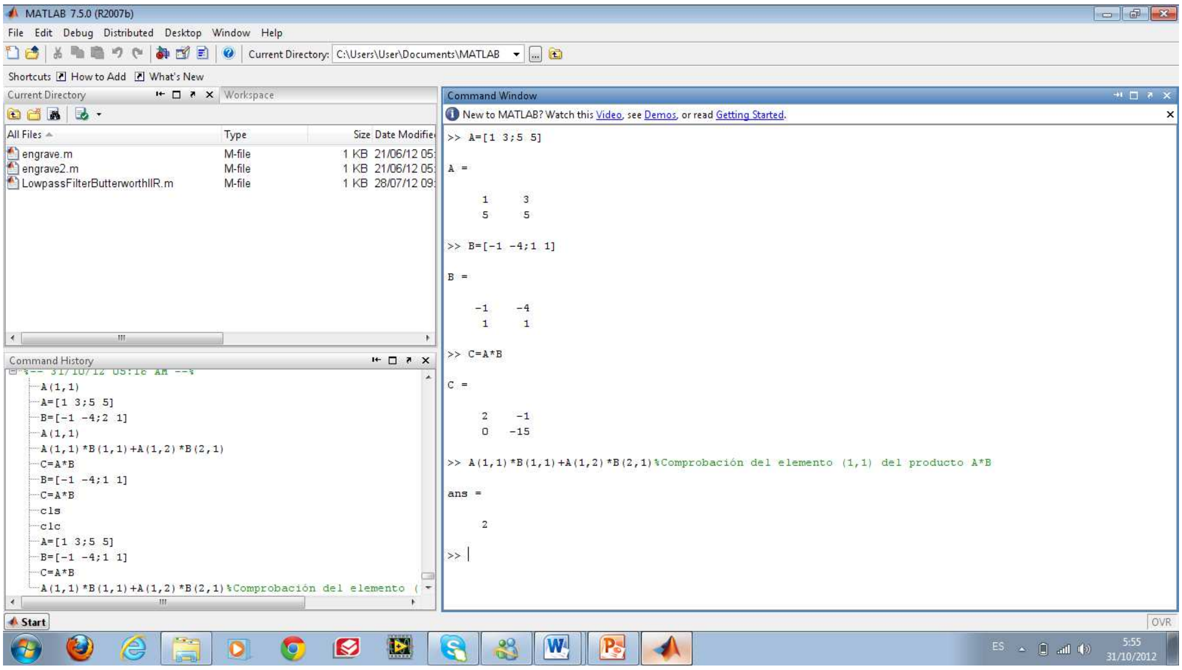


Figura 14.- Comprobación de la matriz producto AB.
Fuente: Autor.

Hay que tener presente que el producto de matrices NO es conmutativo

A*B – B*A

MATLAB interpreta A/B como el producto de *A* por la inversa de *B*. Probar

A/B

A*inv(B)

A*inv(A)

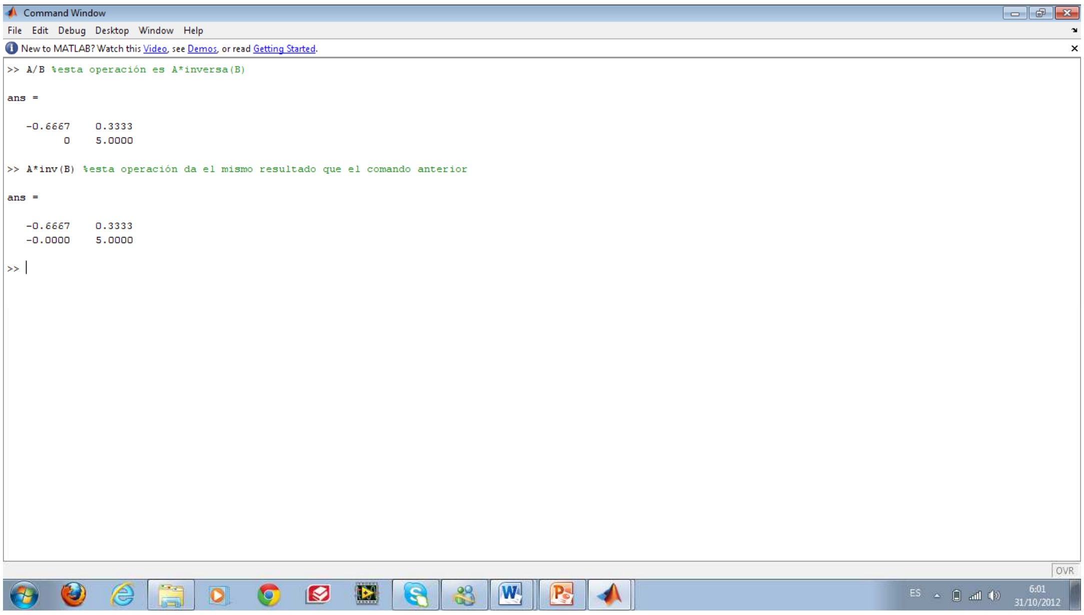


Figura 15.- Comprobación de la matriz A/B.
Fuente: Autor.

La no conmutividad del producto justifica que *inv*(*A*).*B* se abrevie A*B*. Asimismo, la solución del sistema *Ax* = *b*, que formalmente es $x=A^{-1}b$, se obtiene en MATLAB con A*b*. Se debe tener claro que la solución no se obtiene calculando la inversa de *A*, sino aplicando métodos numéricamente más eficientes.

Para multiplicar dos matrices elemento a elemento, en lugar de filas por columnas, se utilizan las variantes ‘punto’ de las operaciones correspondientes. Comprobar

A*B, A.*B
A^ –1, A. ^ –1

Si *A* es una matriz real, *A*’ es la traspuesta de *A*. En el caso complejo, *A*’ es la traspuesta conjugada. La traspuesta sin conjugar se obtiene con *A.*’.

Las instrucciones *fliplr* y *flipud* voltean la matriz de derecha a izquierda y de arriba abajo, respectivamente.

```
A,  flipud(A)
A =
        1       2
        3       4
ans =
        3       4
        1       2

v=1:4,  W=fliplr(v)
v =     1       2       3       4
W =     4       3       2       1
```

3.13. FUNCIONES DE MATLAB.-

MATLAB reconoce las funciones matemáticas elementales:

- Trigonométricas: seno, coseno, tangente, cosecante, secante y cotangente.
- Trigonométricas inversas: arco seno, arco coseno, arco tangente,…….
- Exponencial y logaritmos neperiano, decimal y en base 2.
- Hiperbólicas: seno hiperbólico, coseno hiperbólico, tangente hiperbólica,...
- Hiperbólicas inversas: argumento seno hiperbólico, coseno, tangente,….
- Raíz cuadrada, parte entera, valor absoluto.

La función *sqrt* realiza la raíz cuadrada de un número; si éste es negativo da como resultado un número complejo.

La orden *abs* calcula el valor absoluto de un número; si éste es negativo devuelve su módulo.

```
sqrt(4)
sqrt(-4)
abs(4)
abs(-4)
abs(3+4*i)
```

Los mandatos *exp*, *log* y *log*10 realizan, respectivamente, la exponencial, el logaritmo neperiano y el logaritmo decimal de un número.

```
exp(1)
log(ans)
log10(10)
```

A continuación se listan las funciones matemáticas, que pueden ser aplicadas tanto en escalares como en matrices. Es primordial subrayar que las funciones trigonométricas están expresadas en radianes.

Función	**Descripción**
abs(x)	Valor absoluto o magnitud de un número complejo.
acos(x)	Inversa del coseno.
angle(x)	Ángulo de un número complejo.
asin(x)	Inversa del seno.
atan(x)	Inversa de la tangente.
ceil(x)	Redondea hacia más infinito.
conj(x)	Complejo conjugado.
cos(x)	Coseno.
exp(x)	Exponencial.
fix(x)	Redondea hacia cero.
floor(x)	Redondea hacia menos infinito.
imag(x)	Parte imaginaria de un número complejo.
log(x)	Logaritmo natural.
log10(x)	Logaritmo decimal.
real(x)	Parte real de un número complejo.
rem(x,y)	Resto después de la división.
round(x)	Redondea hacia el entero más próximo.
sign(x)	Devuelve el signo del argumento.
sin(x)	Seno.
sqrt(x)	Raíz cuadrada.
tan(x)	Tangente.

Tabla 5.- Funciones matemáticas de MATLAB.

CAPÍTULO 4

4. CÁLCULOS NUMÉRICOS EN MATLAB

4.1. POLINOMIOS.-

Los polinomios constituyen una clase de funciones de gran importancia, debido a que son fáciles de evaluar y, sobre todo, a que toda función continua en un intervalo cerrado puede aproximarse mediante polinomios.

Los polinomios surgen a menudo en problemas de resolución de ecuaciones. Curiosamente, a pesar de la sencillez y las buenas propiedades de los polinomios, solamente se pueden resolver mediante fórmulas exactas las ecuaciones polinómicas de grados no superior a 4. De hecho, las fórmulas para ecuaciones de grado superior a 2 son raramente utilizadas; en su lugar se utilizan métodos numéricos.

En MATLAB se trabaja con los polinomios de forma sencilla, tan sólo hay que recordar que un polinomio no es nada más que un vector en que el orden de los coeficientes va de mayor a menor.

```
>> p=[3    5    2    8    6]     % 3*x^4+5*x^3+2*x^2+8*x+6
p =
     3    5    2    8    6

>> q=[6    2    1    7    8]     % 6*x^4+2*x^3+x^2+7*x+8
q =
     6    2    1    7    8
```

Además, MATLAB incluye funciones específicas para operar con polinomios. Por ejemplo, si se quiere evaluar lo que vale un polinomio en un punto

```
>> polyval(p,-1)      % Evaluación de 3*x^4+5*x^3+2*x^2+8*x+6 en x=-1
ans =
     -2
```

También es posible multiplicar dos polinomios, usando la función *conv*.

```
>> conv(p,q) % producto de p por q
ans =
     1     36     25     78     113    74     78     106    48
```

U obtener el cociente que se obtiene al dividirlos, recurriendo a la función *deconv*:

```
>> deconv(p,q)      % Cociente resultado de dividir p entre q
ans =
     0.5000
```

```
>> roots(p)    % Raíces del polinomio p
ans =
     -1.7793
     0.4292       +      1.1502i
     0.4292       -      1.1502i
     -0.7458
```

Algunas órdenes relativas a polinomios son las siguientes:

- roots(p) Devuelve las raíces del polinomio *p* en un vector columna.
- poly(r) Construye el polinomio mónico (polinomio en que el coeficiente del término de mayor grado es uno), cuyas raíces se especifican en el vector columna *r*.

- conv(p,q) Multiplica los polinomios *p* y *q*. Para multiplicar más de dos polinomios se repite la orden *conv* tantas veces como sea necesario.
- [c   r]=deconv(d,D) Divide el polinomio *D* por el polinomio *d*. El cociente de la división se guarda en *c* y el resto en *r*.
- polyder(p) Deriva el polinomio *p*.
- polyder(p,q) Deriva el producto *pq* de los polinomios *p* y *q*.
- [a   b]=polyder(p,q) Deriva el cociente *p*/*q* de los polinomios *p* y *q*, guarda el numerador en *a* y el denominador en *b*.
- polyval(p,x) Evalúa el polinomio *p* en cada valor del vector *x*.

MATLAB no dispone de alguna función para sumar o restar polinomios. Se suman como vectores si son del mismo grado, es decir, si los vectores que los representan tienen el mismo número de coordenadas. Si se quieren sumar dos polinomios de distinto grado hay que añadir los ceros necesarios en las primeras coordenadas del vector que representa al polinomio de menor grado, hasta igualar el número de coordenadas de ambos.

4.2. RESOLUCIÓN DE ECUACIONES.-

Para determinar las raíces de una ecuación $f(x) = 0$, se dispone de la orden *fzero* cuya sintaxis es:

- fzero('f',x0) Calcula una raíz de la ecuación $f(x) = 0$ próxima al punto x_0. *f* es una función predefinida de MATLAB, una función definida en un fichero *m* o una cadena de caracteres sin igualar a cero.
- fzero('f',x0,tol) Realiza el mismo cálculo con la tolerancia indicada por el parámetro opcional *tol*, cuyo valor por defecto es *eps*.
- fzero('f',x0,tol,traza) Si el parámetro opcional *traza* es distinto de cero, muestra información en cada iteración.

Ejemplo: Calcular las raíces de la ecuación $x - e^{-x} = 0$.

Solución: Para conocer el número y valor aproximado de las raíces se dibuja la gráfica de la función $x - e^{-x} = 0$ y se le añade una malla.

>> fplot('x-exp(-x)', [-3   3]), grid on $\boxed{\ \hookleftarrow\ }$.

Se observa que hay una raíz próxima a cero. Para calcular su valor se utiliza la orden *fzero*:

>> fzero('x-exp(-x)',0) $\boxed{\ \hookleftarrow\ }$.

4.3. DERIVACIÓN NUMÉRICA.-

Es un método que sirve para evaluar las derivadas de una función en aquellos casos en los que la derivada es difícil de obtener por un método directo, o si se cuenta con los valores funcionales en puntos de datos discretos. MATLAB calcula la derivada mediante el comando *diff*, cuya sintaxis es:

diff(x)

Si x es el vector $(x_1, x_2, \ldots\ldots, x_n)$ genera el vector de $n - 1$ coordenadas $(x_2 - x_1, x_3 - x_2, \ldots\ldots, x_n - x_{n-1})$. Si x es una matiz, el operador de diferencias se aplica a cada columna.

Si se conocen los valores $y_1, y_2, \ldots\ldots, y_n$ de una función $y=f(x)$ en los puntos $x_1, x_2, \ldots\ldots, x_n$, se puede obtener una estimación grosera de los valores de la derivada de f en los puntos $x_1, x_2, \ldots\ldots, x_{n-1}$ mediante el cociente de diferencias

$$f'(x_i) \simeq \frac{y_i+1 - y_i}{x_i+1 - x_i}, \quad i = 1, 2, \ldots, n-1$$

Utilizando la orden *diff*, esto se puede obtener mediante

diff(y)./diff(x)

donde $x = (x_1, x_2, \ldots\ldots, x_n)$ e $y = (y_1, y_2, \ldots\ldots, y_n)$.

EJEMPLO 1.- De una función $y = f(x)$ se conoce la tabla de valores

x	0	1	2	3	4	5
y	1.5	0.8	1.7	4.7	7.2	15

Calcular numéricamente la derivada de la función *f* en los puntos especificados, utilizando la orden *diff*.

Solución:

```
clear, clc
x=0:5;y=[1.5,0.8,1.7,4.7,7.2,15];
der=diff(y)./diff(x);
```

EJEMPLO 2.- Calcular dy/dx para $y = (\cos x)^x$.

La derivada corresponde a la función:

$$dy/dx = y' = y[\ln(\cos x) - x \tan x]; \qquad (1.2)$$

Aplicando el comando *diff* para x en el rango $0 \leq x \leq \pi/2$:

```
% Diferenciación de funciones.
clear
clc
% Rango de x
x=linspace(0,pi/2,1000);
% Cálculos
y=(cos(x)).^x;
dy1=y.*(log(cos(x))-x.*tan(x));
dy2=diff(y)./diff(x);
% Resultados
fprintf('x        dy/dx, exacto          dy/dx, aprox\n\n');
n=length(x)-1;
for i=1:100:n
        fprintf('%6,.4f',x(i));
        fprintf('%12.4f',dy1(i));
        fprintf('%15.4f\n',dy2(i));
end
plot(x,dy1,x(1:50:n),dy2(1:50:n),'*')
title('Diferenciación numérica de y = cos(x)^x')
xlabel('x')
ylabel('dy/dx')
legend('Comando diff')
```

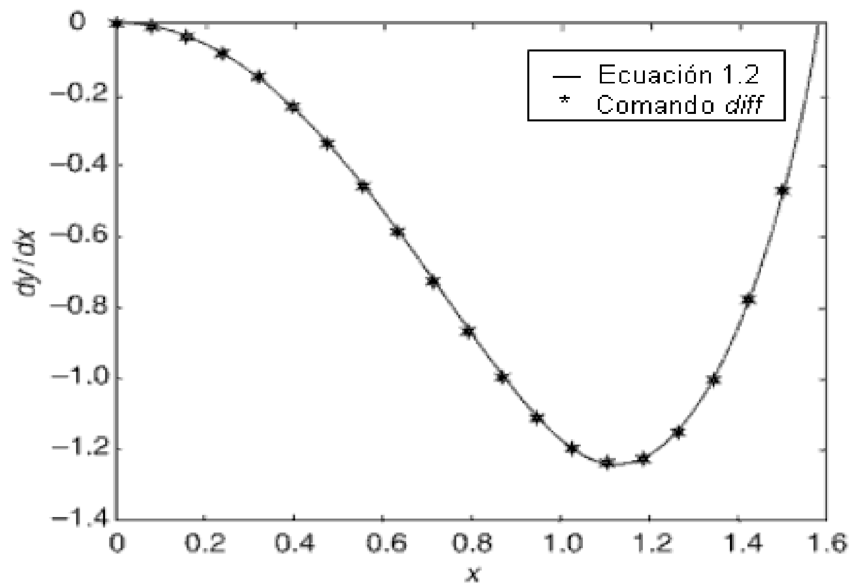


Figura 16.- Diferenciación numérica de $y = (\cos x)^x$
Fuente: Rubén Darío Osorio Giraldo. *Métodos numéricos en química con Matlab*.

4.4. MÁXIMOS Y MÍNIMOS.-

Para calcular los mínimos de una función *f*, MATLAB utiliza la función *fmin*('f',x1,x2). Esta función determina un punto del intervalo $x_1 \leq x \leq x_2$ donde la función indicada alcanza un mínimo relativo.

El argumento función puede ser una función predefinida de MATLAB, una función definida en un fichero *m* o una cadena de caracteres que defina una función. Esta orden admite también argumentos opcionales como la forma [x,valmin]=fmin('f',x1,x2) que da como resultado un vector donde la primera componente es el valor que minimiza la función y la segunda componente es el valor del mínimo.

Puesto que los máximos de una función *f* son los mínimos de –*f* , la orden anterior también permite calcular los máximos de una función.

4.5. CÁLCULOS EN ÁLGEBRA LINEAL.-

Para sistemas de ecuaciones lineales no muy grandes es muy fácil recurrir a su solución con MATLAB.

En MATLAB la multiplicación matricial se denota con el signo *. La función **inv(A)** calcula la inversa de la matriz A.

Cuando hay más ecuaciones que incógnitas (caso sobredeterminado), MATLAB utiliza el operador de división \ ó / automáticamente encuentra la selección que minimiza el error al cuadrado en Ax-b=0. Esta solución se llama *solución de mínimos cuadrados*.

Cuando hay menos ecuaciones que incógnitas (caso indeterminado), existe un número infinito de soluciones. MATLAB calcula dos de forma directa. El uso del operador de división da una solución que tiene ceros para algunos de los elementos de *x*. Alternativamente, calculando *x=pinv(A)*b* se obtiene una solución donde la longitud o norma euclídea de *x* es más pequeña que todas las otras posibles soluciones. Esta solución se llama *solución de norma mínima*.

Otras características se listan a continuación:

A.’	Es la transpuesta de la matriz **A**. La transpuesta compleja conjugada de la matriz **A** se escribe como **A’**.
d=eig(A)	Devuelve los valores propios asociados con la matriz cuadrada A como un vector columna
[V,D]=eig(A)	Devuelve los vectores propios en la matriz **V** y los valores propios como elementos diagonales en la matriz **D**.
[L,U]=lu(A)	Calcula la factorización **LU** de la matriz cuadrada **A**.
[Q,R]=qr(A)	Calcula la factorización **QR** de la matriz **A**.
[U,S,V]=svd(A)	Calcula la descomposición en valores singulares de la matriz **A**.

rank(A)	Devuelve el rango de la matriz **A**.
cond(A)	Devuelve el número de condición de la matriz **A**.
norm(A)	Calcula la norma de la matriz **A**. Admite el cálculo de norma-1, norma-2, norma-F y norma-*x*.
poly(A)	Encuentra el polinomio característico asociado con la matriz cuadrada **A**.
polyvalm(v,A)	Evalúa el polinomio característico **v** usando la matriz cuadrada **A**.

Tabla 6.- Otras funciones características de MATLAB.

4.6. RESOLUCIÓN DE ECUACIONES DIFERENCIALES.-

MATLAB permite resolver ecuaciones diferenciales de dos formas: numérica y simbólica.

Cuando las ecuaciones diferenciales ordinarias se pueden resolver analíticamente, se pueden utilizar características del *toolboox de matemática simbólica* para encontrar soluciones exactas.

Si no se pueden resolver directamente de forma analítica, es conveniente resolverlas numéricamente.

4.6.1. RESOLUCIÓN NUMÉRICA DE ECUACIONES DIFERENCIALES ORDINARIAS.-

MATLAB dispone de las funciones ode23 y ode45 para resolver ecuaciones diferenciales.

Cuando se resuelve de manera simbólica un problema de valor inicial del tipo

$$\begin{cases} u' &= f(t,u), \qquad t \in [T_I, T_F], \\ u(T_I) &= u_I, \end{cases}$$

se obtiene como solución una función *u*(.) que puede ser evaluada en cualquier instante *t* del intervalo [TI, TF]. En cambio, cuando el problema se resuelve mediante un método numérico, sólo es posible obtener una colección finita {u1, u2, …., uN} de aproximaciones de la solución en ciertos instantes t1 < t2 < : : : ;tN, con TI = t1 y TF = tN , es decir,

$$u_n \simeq u(t_n), \quad n = 1, 2, \ldots, N.$$

A grandes rasgos, se puede decir que el error $e_n = u(t_n) - u_n$ (la diferencia entre el valor exacto, que es desconocido, en el instante t_n y el valor aproximado, que es el conocido) depende en gran medida del método numérico empleado.

Además de los métodos numéricos para resolver problemas de valor inicial, mencionados anteriormente, existe una amplia variedad de la que se recalcan dos de ellos:

`[t,u] = ode45(FUN,[TI TF],uI)`	resuelve numéricamente el sistema diferencial de ecuaciones `u' = f(t,u)` en el intervalo `[TI TF]`, con la condición inicial `u(TI)=uI`. `FUN` es una función de MATLAB tal que `FUN(t,u)` es un vector columna con el valor de `f(t,u)`. El método numérico es el par RK5(4) de Dormand y Prince.
`[t,u] = ode15s(FUN,[TI TF],uI)`	Análogo al anterior, la diferencia está en el tipo de método numérico empleado.

En ambos casos, se obtiene como respuesta un vector columna *t* que contiene los instantes en los que se ha calculado la solución aproximada, y el vector *u* -o una matriz, en el caso de un sistema- que contiene tales aproximaciones.

Los instantes contenidos en *t* son elegidos automáticamente por el método numérico; si se desean instantes predeterminados, basta reemplazar el argumento [TI TF] por [T1 T2 ... TN].

La solución aproximada se puede representar gráficamente con la orden *plot*.

EJEMPLO 1.- Para resolver numéricamente el problema

$$\begin{cases} u' &= (0,7 - 0,01 \cdot u)u, \qquad t \in [0,10], \\ u(0) &= 20. \end{cases}$$

basta escribir el archivo de función

f.m
```
function du = f(t,u)
du = (0.7 - 0.01 * u) * u ;
```

y luego ejecutar la orden

>> [t,u] = ode45('f',[0 10],20);

Como respuesta, se obtienen dos vectores columna *t* y *u* cuyos comienzo y fin son

```
t = [ 0        0.1005  0.2010  0.3014  ...   9.7010  9.8505 10.0000]
u = [20.0000 21.0197 22.0684 23.1448  ...  69.8038 69.8232 69.8408]
```

EJEMPLO 2.- Para resolver numéricamente el problema del ejemplo anterior en el intervalo [0; 1], la función f puede ser

f.m
```
function du = f(t,u)
du1 = 1/4 * u(1) + 3/4 * u(2);
du2 = 9/4 * u(1) - 5/4 * u(2);
du = [du1;du2];
```

y luego bastará ejecutar la orden

```
>> [t,u] = ode45('f',[0 1],[0.2; 0.05]);
```

La variable de salida *u* es ahora una matriz de dos columnas: en la primera están los valores aproximados de u1(.), y en la segunda columna están los de u2(.):

```
     t            u
  ------    -------------
       0    0.2000  0.0500
  0.0065    0.2006  0.0525
  0.0130    0.2012  0.0550
    ⋮         ⋮       ⋮
  0.9630    0.4311  0.4093
  0.9815    0.4389  0.4178
  1.0000    0.4468  0.4265
```

4.6.2. RESOLUCIÓN SIMBÓLICA DE ECUACIONES DIFERENCIALES ORDINARIAS.-

Para resolver de forma exacta o simbólica una o varias ecuaciones diferenciales, MATLAB dispone de la orden *dsolve*.

Por defecto, la variable independiente es *t*, pero se puede usar cualquier otra variable si se incluye como último argumento:

`dsolve('ec1',...,'ecn')`	resuelve el sistema diferencial de ecuaciones y condiciones iniciales {`ec1,...,ecn`}
`dsolve('ec1',...,'ecn','var')`	resuelve el sistema diferencial de ecuaciones y condiciones iniciales {`ec1,...,ecn`}, y usa `var` como variable independiente

El operador *D* se utiliza para representar la derivación con respecto a la variable independiente, es decir, *u'* se escribe *Du*; las derivadas de orden superior *u''*, *u'''*,…. se escriben *D2u*, *D3u*,…., cuando se resuelve un sistema de ecuaciones diferenciales, MATLAB proporciona las funciones solución en orden léxico-gráfico. Por ejemplo, la ecuación diferencial lineal

$$a_2 \frac{d^2x}{dt^2} + a_1 \frac{dx}{dt} + a_0 x = b$$

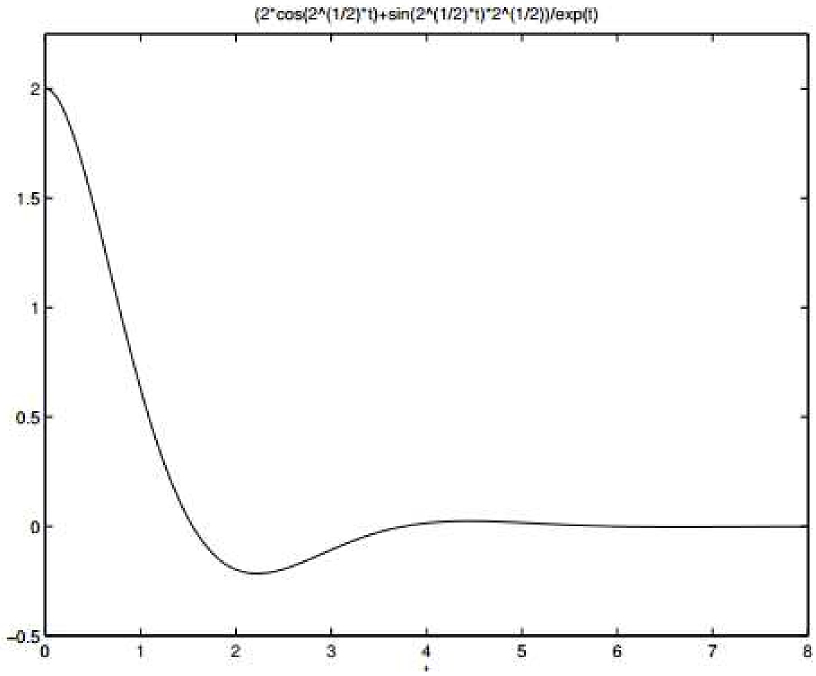


Figura 17.- Derivación de una ecuación diferencial.
Fuente: J.M. González de Durana, *Introducción a MATLAB*,

EJEMPLO 1.- Para resolver el problema de valores iniciales

$$u' = \frac{1}{2}u, \qquad u(0) = \frac{1}{4},$$

basta escribir

```
>> u = dsolve('Du = u/2','u(0) = 1/4')
```

Se obtiene

u =

1/4*exp(1/2*t)

es decir, la solución es

$$u(t) = \frac{1}{4}e^{t/2}$$

La solución se puede representar gráficamente utilizando *ezplot*; por ejemplo, en el intervalo [0,3].

>> ezplot(u,[0 3])

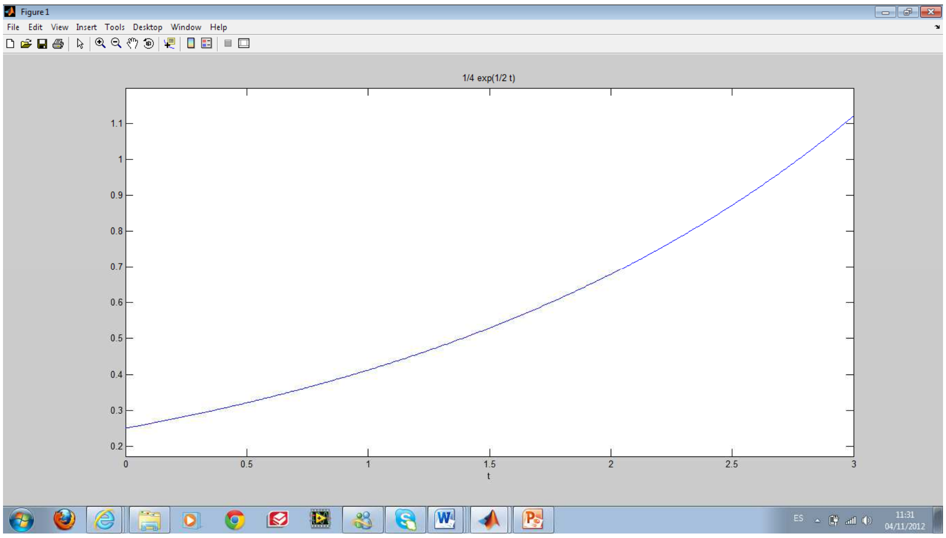


Figura 18.- Representación gráfica del problema de valores iniciales del Ejemplo 1.
Fuente: Autor.

EJEMPLO 2.- Es posible obtener la solución general de una ecuación diferencial:

>> u = dsolve('Du=u/2')

u =

C1*exp(1/2*t)

Esto significa que la solución de la ecuación diferencial u' = u=2 es

$$u(t) = C_1\, e^{t/2}, \qquad (C_1 \in \mathbb{R})$$

EJEMPLO 3.- La ecuación diferencial, o la condición inicial, pueden contener parámetros. Por ejemplo, si se desea resolver el problema

$$\begin{cases} u' &= r \cdot u, \qquad t \geq 0, \\ u(0) &= u_0. \end{cases}$$

Se ingresa

```
>> u = dsolve('Du = r*u','u(0) = u0')
```

y se obtiene

```
u =
        u0*exp(r*t)
```

4.6.2.1. TRANSFORMADA DE LAPLACE.-

En los sistemas de control con modelo matemático de función de transferencia se presenta el problema del cálculo de la transformada inversa, de cara a la obtención de la respuesta del sistema (variable *y*(*t*) de salida) dada la variable de entrada *x*(*t*). Las variables de entrada y de salida son funciones del tiempo.

X(*s*) e *Y*(*s*) son las transformadas de Laplace de *x*(*t*) e *y*(*t*) respectivamente. Por tanto, dada x(t),

$$\left.\begin{array}{l} X(s) = \mathcal{L}[x(t)] \\ Y(s) = X(s)G(s) \end{array}\right\} \Longrightarrow y(t) = \mathcal{L}^{-1}[Y(s)]$$

$X(s)$, $Y(s)$ y $G(s)$ son funciones racionales.

a) RESOLUCIÓN SIMBÓLICA.-

La librería *Symbolic Toolbox* de MATLAB permite hallar, mediante las órdenes **laplace** e **ilaplace**, las transformadas directa e inversa de Laplace. Dada una función $f(t)$, se denota por $F(s)$ su transformada de Laplace. Es decir

$$\mathcal{L}(f(t)) = F(s).$$

Para hallar la transformada de Laplace $F(s)$ de la función $f(t)$ se ingresa

```
>> F = laplace(f)
```

en donde *f* es la expresión simbólica de $f(t)$. Por defecto MATLAB asume que la variable independiente de *f* es *t*. Para hallar la transformada inversa $f(t)$ de una $F(s)$, se pone

```
>> f = ilaplace(F)
```

en donde *f* es la expresión simbólica de $F(s)$. Por defecto MATLAB asume que la variable independiente de *F* es *s*.

Como ejercicio, se resolverá el siguiente ejemplo:

```
>> syms s m b k
>> G = 1/(m*s^2+b*s+k)
>> G1 = subs(G,[m,b,k],[1,1,1])
G1 =
1/(s^2+s+1)
```

```
>> Y = symmul(G1,1/s);
>> y = ilaplace(Y);
>> ezplot(y, [0,15]), axis([0, 15, 0, 1.25]), title('y(t)')
```

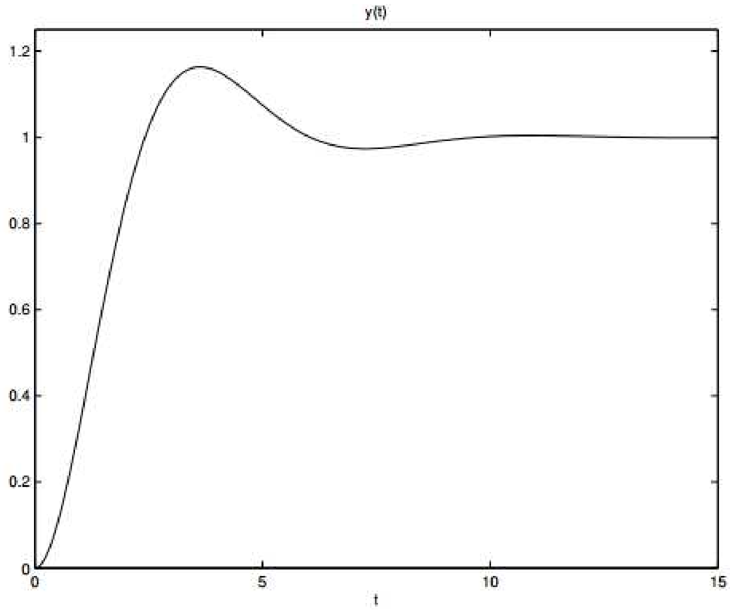


Figura 19.- Representación de la transformada de Laplace.
Fuente: J.M. González de Durana, *Introducción a MATLAB*.

4.7. ECUACIONES DIFERENCIALES DE PRIMER ORDEN.-

Una ecuación diferencial se la escribe siempre entre comillas simples. En MATLAB, *y′* se representa por *Dy*; *y″* se representa por *D2y*; *y‴* por *D3y*, etc.

Por ejemplo, la ecuación $\frac{d^3y}{dt^3} + 4\frac{dy}{dt} = \operatorname{sen}^2 t$ se escribiría en MATLAB como ’D3y+4*Dy=sin(t)^2’.

Las condiciones iniciales también van entre comillas simples. Por ejemplo, las condiciones y(0) = 1, y′(0) = 2, y″(0) = 3 se escriben ′y(0) = 1′, ′Dy(0) = 2′, ′D2y(0) = 3′.

Orden	Descripción
dsolve(ecuación,'x')	Devuelve la solución general de una ecuación diferencial respecto de la variable independiente *x*.
dsolve(ecuación,condicion1,condicion2,....,'x')	Devuelve la solución de la ecuación diferencial respecto de la variable independiente *x* verificando las condiciones iniciales indicadas.

4.8. SISTEMAS DE ECUACIONES DIFERENCIALES DE PRIMER ORDEN.-

Un sistema de ecuaciones diferenciales de primer orden puede escribirse como

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}), \qquad \mathbf{y}(t_0) = \mathbf{y}_0$$

donde y', y e y_0 son vectores con el mismo número de componentes. Para resolver este tipo de problemas se utilizan los mismos métodos que para resolver la ecuación escalar. La única diferencia desde el punto de vista de MATLAB es que la única forma de definir la función vectorial *f* es mediante un fichero de función (fichero *.m*) que ha de devolver además un vector columna.

Orden	Descripción
dsolve(ec1,ec2,....,'x')	Devuelve la solución general del sistema de ecuaciones diferenciales de variable independiente *x*.
dsolve(ec1,ec2,....,cond1,cond2,....,'x')	Devuelve la solución del sistema respecto de la variable independiente *x* verificando las condiciones iniciales indicadas.

EJEMPLO 1.- Hallar la solución del problema de valores iniciales

$$\begin{cases} u' = u + 2v \\ v' = -2u + v + 2e^x \end{cases} \quad u(0) = 2,\ v(0) = 0$$

Utilizando **dsolve**, se escribiría

```
>> syms x, ed1='Du=u+2*v'; ed2='Dv=-2*u+v+2*exp(x)';
S= dsolve(ed1,ed2,'u(0)=2','v(0)=0','x')
S =
        v: [1x1 sym]
        u: [1x1 sym]
```

y esto indica que MATLAB ha calculado las soluciones y las ha almacenado en la estructura *S*. De este modo, para definirlas bastará escribir:

```
>> u=S.u,v=S.v
u =
        exp(x) + cos(2*x)*exp(x)
v =
        -sin(2*x)*exp(x)
```

También en este caso, si se omite la variable, el programa interpreta que se trabaja con variable independiente *t*.

EJEMPLO 2.- Obtener la solución en el intervalo [0, 10], del sistema

$$\begin{aligned} x' &= y - z \\ y' &= z - x \\ z' &= x - y \end{aligned}$$

Con las condiciones iniciales $x(0) = -1$, $y(0) = 0$ y $z(0) = 1$. Este problema puede escribirse también

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \mathbf{f}\left(t, \begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} y - z \\ z - x \\ x - y \end{bmatrix}, \qquad \begin{bmatrix} x(0) \\ y(0) \\ z(0) \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

Para resolverlo en el intervalo [0, 10] primero habría que definir la función de un fichero

```
function dy = sistema(t, y)
dy = zeros(3,1);      %vector columna
dy(1) = y(2) – y(3);
dy(2) = y(3) – y(1);
dy(3) = y(1) – y(2);
```

aunque *t* no aparezca en ninguna de las ecuaciones del sistema debe ser un dato del fichero; la segunda línea fuerza a que la función devuelva un vector columna, tal como requieren las funciones **ode***, por último, nótese que aunque en el sistema se hayan utilizado diferentes letras para las variables dependientes, tal como es usual, en MATLAB hay que trabajar con un vector, por tanto se utiliza un solo nombre (*y*) para todas las variables y estas se distinguen porque corresponden a distintos elementos del vector, de forma que la traducción es más natural si se escribe el sistema como

$$\begin{aligned} y_1' &= y_2 - y_3 \\ y_2' &= y_3 - y_1 \\ y_3' &= y_1 - y_2 \end{aligned}$$

Una vez escrita la función se obtiene la solución con la orden

```
[T, Y] = ode*(@sistema, [0   10], [-1   0   1]);
```

La solución puede verse gráficamente con la orden ode*(@sistema, [0 10], [-1   0   1]);

La matriz *Y* tiene tres columnas. La primera corresponde a los valores de la variable *x*, la segunda a la *y*; y la tercera a la variable *z*.

4.9. ECUACIONES DIFERENCIALES DE ORDEN SUPERIOR.-

Una ecuación diferencial ordinaria de orden superior es una expresión que relaciona una variable dependiente *y*, y sus derivadas de cualquier orden con respecto a una variable independiente *x*, así:

$$f\left(x, y(x), Dy(x), D^2y(x), \ldots, D^n y(x)\right) = r(x)$$

En donde $Dy(x), D^2y(x), \ldots, D^n y(x)$ son las derivadas de orden 1, 2, ..., *n* de la función *y*(*x*).

Por analogía con las ecuaciones diferenciales de primer orden, una solución general de la ecuación diferencial es una familia de curvas del plano que contiene *n* constantes arbitrarias, así:

$$F(x, y, C_1, C_2, \ldots, C_n)$$

Son ejemplos de ecuaciones diferenciales de orden superior, las siguientes:

1. $y''(x) - xy(x) = 0$
2. $a_2 y''(t) + a_1 y'(t) + a_0 y(t) = f(t)$

3. $y''(t) + 4\sin(y(t)) = 0$

4. $x^3y'''(x) + \alpha x^2y''(x) + \beta xy'(x) + \gamma y(x) = f(x)$

5. $x^2y''(x) + xy'(x) + (x^2 - \gamma^2)y(x) = 0$

De las ecuaciones mostradas, la tercera es no lineal y el resto son lineales. La segunda ecuación es de coeficientes constantes y recibe el nombre de ecuación de oscilaciones.

La primera ecuación es la ecuación de Airy. La cuarta es la ecuación diferencial de Euler de tercer orden y la última es la ecuación diferencial de Bessel.

EJEMPLO 1.- Resolver el problema

$$y'' - 4y' + 3y = 9x^2 + 4,\ y(0) = 6,\ y'(0) = 8$$

```
>> y=dsolve('D2y-4*Dy+3*y=9*x^2+4','y(0)=6','Dy(0)=8','x')
y =
        2*exp(3*x)-6*exp(x)+10+8*x+3*x^2
```

Si se omite la variable, el programa interpreta que se trabaja con variable independiente *t.*

EJEMPLO 2.- Resolver la ecuación

$$\frac{d^3y}{dt^3} + 4\frac{dy}{dt} = \text{sen}^2 t$$

```
>> clear, syms t
y=dsolve('D3y+4*Dy=sin(t)^2')
y =
        -1/2*cos(2*t)*C2+1/2*sin(2*t)*C1-3/64*sin(2*t)+1/16*cos(2*t)*t+1/8*t+C3
```

CAPÍTULO 5

5. METODOLOGÍA USADA Y PRÁCTICAS DE LABORATORIO RECOMENDADAS

5.1. METODOLOGÍA.-

Para alcanzar los objetivos propuestos se seleccionaron los siguientes métodos de investigación:

- Método de análisis.
- Método de Organización.
- Método de investigación acción y Exploratoria.
- Métodos de comprobación y de observación (pre-Experimental).

5.2. JUSTIFICACIÓN DE LA ELECCIÓN DEL MÉTODO.-

MATLAB es una herramienta muy versátil utilizada en múltiples aplicaciones científicas y tecnológicas, por lo que resulta idóneo utilizar este programa como apoyo a las clases de Cálculo en la Facultad de Ciencias Técnicas para el Desarrollo de la UCSG.

Puesto que MATLAB es una herramienta potente para resolver problemas elementales (como números reales y complejos, vectores y matrices), además de otros problemas más complejos, gracias a las funciones predefinidas, es posible encontrar de manera sencilla las soluciones de límites, derivadas e integrales, representación gráfica, entre otros, todo esto ejecutando sencillas órdenes.

5.3. PROCEDIMIENTOS A SEGUIR PARA LA REALIZACIÓN DE LAS PRÁCTICAS.-

Adicional a la investigación, selección y análisis de varios textos, apuntes y documentos virtuales de las distintas temáticas que componen el presente trabajo, se procedió a las prácticas de laboratorio aplicando los métodos que se proponen y se explican en cada uno de ellos con el programa MATLAB y así confrontar las soluciones manuales con las realizadas en el sistema.

5.4. PRÁCTICAS BÁSICAS REALIZADAS.-

Sustentado en el método seleccionado de comprobación y de observación (pre-Experimental), para realizar las prácticas se partió del modelo experimental analítico con una perspectiva mesurable con procedimientos de los fenómenos físicos y científicos basados en la realidad, observando y comprobando los diferentes casos para los que son aplicables las ecuaciones o sistemas de ecuaciones diferenciales.

6. CONCLUSIONES

La materia Cálculo Tres es elemental en la carrera de Ciencias Técnicas para el Desarrollo de la UCSG , así como en otras profesiones, por lo que es importante su comprensión y aprendizaje, ya que es el fundamento de varias materias en los siguientes semestres, tal como Teoría de Control I, Teoría de Control II, entre otras.

El estudio del programa MATLAB se vuelve importante en el estudiante de las carreras técnicas de la Facultad de Educación Técnica para el Desarrollo en la Universidad Católica Santiago de Guayaquil.

MATLAB es una potente herramienta que simplifica la resolución de ecuaciones diferenciales de primer grado mediante la utilización de comandos adecuados, no sólo la solución analítica sino también la solución gráfica tanto en 2 y 3 Dimensiones.

MATLAB se ha convertido en una herramienta eficaz en los ámbitos educativo e industrial. En el área educativa representa una herramienta principal para el estudio de matrices, resolución de ecuaciones en nivel básico y avanzado. A nivel industrial es altamente aplicado en problemas de ingeniería y matemáticas.

Utilizando los ficheros *M* es posible la simulación y entendimiento del uso de diversos procesos matemáticos complejos que resultan difíciles de analizar utilizando ecuaciones pero con la importante ayuda de MATLAB se simplifica esta tarea.

Después de investigar, experimentar y evaluar las herramientas de las que dispone MATLAB, se ha obtenido un material de soporte para la materia Cálculo.

7. RECOMENDACIONES

Practicar distintos ejemplos y realizar todos los ejercicios propuestos, ya que la única manera de aprender a utilizar cualquier programa es manejándolo e intentando hacer lo que se requiere.

Presentar nuevas opciones de simulación y nuevas prácticas asesoradas por cada uno de los catedráticos de Cálculo, utilizando MATLAB.

Incorporar más horas de prácticas de laboratorio en la carrera desde los primeros años de estudio como complemento curricular.

Integrar el MATLAB como herramienta en otras materias y así abarcar una mayor área de desarrollo de aplicaciones y de este modo contar con un instrumento bastante útil, eficiente y eficaz para la modelación numérica, desarrollo y gráfico de distintos problemas.

8. BIBLIOGRAFÍA

- ARANDA, Tomás; GARCÍA, J. Gabriel. *Notas sobre MATLAB*. Textos Universitarios EDIUNO. Ediciones de la Universidad de Oviedo. 1999.
- CÁNOVAS PEÑA, José S. *Apuntes de ecuaciones diferenciales*. Mayo 2004.
- CAMPOS SANCHO, Beatriz; CHIRALT MONLEON, Cristina. *Ecuaciones diferenciales*. Primera edición. Collecció Sapientia, 49. Publicacions de la Universitat Jaume I. Servei de Comunicació i Publicacions. Campus del Riu Sec. Edifici Rectorat i Serveis Centrals. 12071 Castelló de la Plana. 2011.
- CHÁVEZ ASTURIAS, Karl Heinz. *MÉTODOS NUMÉRICOS PARA RESOLVER PROBLEMAS DE ECUACIONES DIFERENCIALES LINEALES CON VALOR INICIAL*. Universidad Francisco Marroquín. Facultad de Ingeniería de Sistemas, Informática y Ciencias de la Computación. Guatemala. 2000.
- CORDERO BARBERO, A.; HUESO PAGOAGA, J.L.; MARTÍNEZ MOLADA, E.; TORREGROSA SÁNCHEZ, J.R. *MÉTODOS NUMÉRICOS CON MATLAB*. Editorial Universidad Politécnica de Valencia. Valencia. 2005.
- EDWARDS, C. Henry; PENNEY, David E. *Ecuaciones diferenciales.* Cuarta Edición. Prentice Hall. Pearson Educación de México, S.A. de C.V. México. 2001.
- FUSTER, R.; GIMÉNEZ, I. *VARIABLE COMPLEJA YE CUACIONES DIFERENCIALES*. Editorial Reverté S.A. 1995.
- GIL RODRÍGUEZ, Manuel. *INTRODUCCIÓN RÁPIDA A MATLAB Y SIMULINK PARA CIENCIA E INGENIERÍA*. Ediciones Díaz de Santos, S.A. Madrid. 2003.
- GILAT, Amos. *MATLAB: UNA INTRODUCCIÓN CON EJEMPLOS PRÁCTICOS*. Editorial Reverté S.A. 2006.

- GONZÁLEZ DE DURANA, J.M., *Introducción a MATLAB*, Dpto. de Ingeniería de Sistemas y Automática. EUITI e ITT, UPV-EHU. VITORIA-GASTEIZ. Enero, 2004.
- JIMÉNEZ LÓPEZ, Víctor. *Ecuaciones Diferenciales: cómo aprenderlas, cómo enseñarlas*. Publicaciones Universidad de Murcia. España. 2000.
- OSORIO GIRALDO, Rubén Darío. *Métodos numéricos en química con Matlab*. Editorial Universidad de Antioquía. Medellín-Colombia. 2007
- PADILLA REYES, Jorge Mario. *MÉTODOS DIRECTOS E ITERATIVOS PARA LA SOLUCIÓN DE SISTEMAS DE ECUACIONES LINEALES*. Universidad Industrial de Santander. Facultad de Ingenierías Físico-Químicas. Escuela de Ingeniería de Petróleos. Bucaramanga-Colombia. 2010.
- PURCELL, Edwin J.; VARBERG, Dale; RIGDON, Steven E. *Cálculo Diferencial e Integral*. Novena Edición. Pearson Educación. México. 2007.
- ROBLES DEL PESO, Arturo; GARCÍA BENEDITO, Julio. *Métodos numéricos en Ingeniería. Prácticas con Matlab.* Textos Universitarios EDIUNO. Ediciones de la Universidad de Oviedo. Asturias. 2006.
- ROMERO BAUSET, José Vicente; ROSELLÓ FERRAGUT, María Dolores; ZALAYA BÁEZ, Ricardo. *FUNDAMENTOS MATEMÁTICOS PARA LA INGENIERÍA CON MATLAB*. Editorial Universidad Politécnica de Valencia. Valencia-España. 2002.
- ZILL, Dennis G.; CULLEN, Michael R. *Ecuaciones Diferenciales, Con Problemas con valores en la frontera*. Séptima Edición. Cengage Learning. México. 2009.

ENLACES:

- http://es.scribd.com/doc/59938413/21/Ecuaciones-Lineales-de-Primer-Orden
- http://deymerg.files.wordpress.com/2011/07/capitulo-2.pdf
- http://www2.uah.es/rviana/Sesion27Amb10-11.pdf
- http://www.ehu.es/izaballa/Ecu_Dif/Apuntes/lec8.pdf
- http://www.reverte.com/catalogo/img/pdfs/9788429151626.pdf
- http://gcp.fcaglp.unlp.edu.ar/_media/integrantes:psantamaria:numerico1:numerico1-06.pdf
- http://campus.usal.es/~mpg/Personales/PersonalMAGL/Docencia/MetNumTema4Teo(09-10).pdf
- http://www.bioingenieria.edu.ar/academica/catedras/control/archivos/material/Anexos/apunte%20matlab%20parte1%20y%202.pdf
- http://books.google.com.ec/books?id=jrHpzQHIOA8C&pg=PA1&lpg=PA1&dq=aritmetica+elemental+matlab&source=bl&ots=ANoHW8mdZ_&sig=ma_eRKVc-rdcRTbsiS8pP9Hn-Xo&hl=es&sa=X&ei=
- http://macabar.webs.upv.es/xelo/custom/metnum/ode.pdf
- http://es.scribd.com/doc/57741542/5/I-2-2-Soluciones-generales-y-particulares-de-una-E-D
- http://mazinger.sisib.uchile.cl/repositorio/ap/ciencias_quimicas_y_farmaceuticas/apmat4d/09a.html