

EL USO DE MATLAB COMO RECURSO DOCENTE PARA LA ENSEÑANZA-APRENDIZAJE DE LOS MÉTODOS NUMÉRICOS

LUIS LORENZO JIMÉNEZ GARCÍA; PROFESOR DE CARRERA FES ARAGÓN-UNAM; jimenezg@att.net.mx

RESUMEN.

El conocimiento de la solución de problemas de ingeniería por medio de *Métodos Numéricos*, en la formación integral de los alumnos de ingeniería de las FES-Aragón, reviste de gran importancia. Es necesario contar con los recursos académicos para hacer más dinámica y atractiva la enseñanza, dándole un enfoque teórico y práctico. Con el desarrollo de las tecnologías de la información y comunicación se han transformado los procesos de enseñanza-aprendizaje, ya que es posible emplear una nueva variable fundamental para la comprensión global de conceptos, que se define como “visualización”, mediante lenguajes de programación aplicados. Existen varios tipos de software como son MatLab, Mathematica, Maple, Mathcad y Derive entre otros. Por experiencia profesional y como recurso de apoyo para el mejoramiento de la docencia se recomienda usar MATLAB por su diversidad de aplicaciones e interacción.

El desarrollo del presente trabajo contempla analizar y aplicar los *Métodos Numéricos* con ayuda de los comandos y funciones de MATLAB. Se resuelven y se presenta los resultados obtenidos de la solución de problemas aplicados a ingeniería. MATLAB permite reforzar los conocimientos de los *Métodos Numéricos*, facilitando a los alumnos de ingeniería la mejor comprensión de estos procesos matemáticos, dando como resultado un mejoramiento de la docencia para un mejor aprendizaje.

INTRODUCCIÓN.

Las Nuevas Tecnologías de la Información y la Comunicación (NTIC) han aportado a la formación en el área de Métodos Numéricos una nueva variable fundamental para la comprensión global de conceptos, que se definen como “visualización”, entendiendo como tal el conjunto de representaciones geométricas o gráficas de conceptos matemáticos, principios o problemas generados por la computadora. De esta manera visualizar hace posible, “...crear ricas imágenes mentales que el individuo pueda manipular en su mente, ensayando diferentes representaciones del concepto matemático y su solución”. Entre las ventajas que supone la utilización de las nuevas tecnologías, cabe destacar el hecho de poder mostrar a los alumnos que los Métodos Numéricos no son sólo un conjunto de definiciones, teoremas, demostraciones y métodos repetitivos, muchas veces de difícil comprensión, sino que están en concordancia con los últimos e innovadores avances informáticos. De este modo podemos ver cómo los resultados de dichos avances pueden, y deben, ser utilizados día a día en el salón de clases, ya que facilitan enormemente los cálculos y permiten, además, visualizar a los alumnos, muchos de los problemas que les son planteados. Contemplar la gráfica de una función, interpretar geoméricamente algunos conceptos o soluciones de problemas no tiene porqué resultar tan tedioso como puede parecer. El software MatLab es un potente lenguaje de programación de cuarta generación. Es un programa interactivo que ayuda a realizar cálculos numéricos y simbólicos, analizando y visualizando los datos, para resolver problemas matemáticos, físicos, etc. La utilización de MatLab como un recurso docente, permite explicar conceptos que, de otra forma, quedarían en un nivel de abstracción difícil de asimilar por muchos alumnos en un tiempo breve: volúmenes generados por funciones al rotar sobre un eje, representaciones de superficies en 2D y 3D, conceptos y resultados teóricos susceptibles de ser comprobados empíricamente y realizar operaciones complejas de cálculo. También permite una participación constructivista por parte del alumno, ya que puede conjeturar, experimentar y extraer conclusiones. MatLab es un potente recurso matemático que acompañará siempre al

alumno en su proceso de aprendizaje, ya que con mínimos conocimientos informáticos ofrece toda una gama de posibilidades para resolver los problemas de Métodos Numéricos, tales como la obtención de raíces de ecuaciones trascendentes y algebraicas, solución de sistemas de ecuaciones lineales y no lineales, obtención de los polinomios de Taylor, interpolación, integración, solución de ecuaciones y sistemas de ecuaciones diferenciales y solución de ecuaciones en derivadas parciales, entre otros.

ANÁLISIS.

El uso de las NTIC se ha utilizado como recurso didáctico en la búsqueda de nuevos métodos de enseñanza-aprendizaje. Estos avances tecnológicos han generado software de aplicación (como MatLab) que hace que sea especialmente interesante reflexionar acerca de cómo esas tecnologías pueden modificar los procesos de enseñanza y aprendizaje de los Métodos Numéricos. Las experiencias de más de 30 años de actividades académica en el área de físico matemáticas y en especial de la impartición de la asignatura de Métodos Numéricos a los estudiantes de las diferentes carreras de ingeniería de FES Aragón, ha demostrado que el uso de MatLab como un recurso didáctico de apoyo en la solución de problemas, propicia y despierta el interés por la parte algorítmica y analítica que contiene los Métodos Numéricos. No se debe olvidar que estas NTIC en sí mismas no promueven el aprendizaje y no constituyen ninguna panacea de carácter universal ni ninguna garantía de eficacia pedagógica, todo dependerá de la opción y concepción pedagógica por la cual se elija diseñar un determinado modelo educativo. Por último, basado en la experiencia, se ha hecho una investigación de corte cuantitativo, cuya fuente de investigación por profundidad es descriptiva. La información se obtuvo en forma experimental, siguiendo una metodología cuantitativa y de investigación comparada.

A continuación se presentan algunos casos de ejercicios que hacen más dinámico y activo el sistema de enseñanza aprendizaje de los Métodos Numéricos, empleando el software MatLab.

La Tabla 1 muestra los comandos y funciones MatLab empleados en el presente trabajo, así como su empleo en los Métodos Numéricos que a continuación se muestran:

Tabla 1: Comandos y funciones para la solución de los diversos Métodos Numéricos

Función	Descripción
fzero(fun,x0)	Encuentra una sola raíz de la función (<i>fun</i>) con una variable, es decir, cero de funciones, cerca del valor aproximado <i>x0</i> , si <i>x0</i> es un escalar. El algoritmo para encontrar las raíces de una función trascendente o polinomial mediante el comando <i>fzero</i> emplea una combinación del método de Bisección de Bolzano, método de la Secante y el método de Interpolación cuadrática inversa.
roots(V)	Encuentra las raíces (reales o complejas, diferentes o iguales) de un polinomio (ecuación algebraica), cuyos coeficientes son los componentes del vector V. El resultado presenta un vector columna cuyos elementos son las raíces del polinomio V. El algoritmo del comando <i>roots</i> para encontrar las raíces está basado en el cálculo de los valores característicos (eigenvalores) asociados a la llamada matriz propia de Fröbenius.

rref([A, B])	Obtiene la matriz reducida escalonada por renglones de A, utilizando el método de Gauss-Jordan, en la cual, la diagonal principal tiene 1 y los demás elementos 0. El número de renglones no nulos de rref(A) es el rango de A. Además, muestra cuando un sistema es incompatible o indeterminado mediante el vector de términos independientes B.
x = fsolve(f, x0)	Resuelve el sistema de ecuaciones no lineales dado en f y con los valores iniciales $X0$ por los Métodos de Gauss-Newton, Levenberg-Marquardt y Gran escala.
quad(f,a,b)	Halla la integral $\int_a^b f(x) dx$ por el método de Simpson con un error de 10^{-6} . La función f usa los comandos <i>inline</i> ó <i>@</i> para declararse. Los límites de la integral a,b deben ser valores escalares.
quadl(f,a,b)	Halla la integral $\int_a^b f(x) dx$ por el método de la cuadratura de Lobatto con un error de 10^{-6} . La función f usa los comandos <i>inline</i> ó <i>@</i> para declararse. Los límites a,b de la integral deben ser valores escalares.
dblquad(f,xmin,xmax, ymin,ymax)	Halla la integral $\int_a^b \int_c^d f(x, y) dx dy$ con un error de 10^{-6} , por el método de Simpson. La función $f(x, y)$ usa los comandos <i>inline</i> ó <i>@</i> para declararse. Los límites de la integral doble deben ser valores escalares.
triplequad(f,xmin,xmax, ymiin, ymax, zmin,zmax)	Halla la integral $\int_a^b \int_c^d \int_e^f f(x, y, z) dx dy dz$ con un error de 10^{-6} , por el método de Simpson. La función $f(x, y, z)$ usa los comandos <i>inline</i> ó <i>@</i> para declararse. Los límites de la integral triple deben ser valores escalares.
ode45	(Ordinary differential equation solver of 4th and 5th order). Es un método de un solo paso, es decir, para obtener un punto, sólo se basa en el calculado anteriormente. Está basado en la fórmula explícita de Runge-Kutta de 4° y 5° orden (Par de Dormand-Prince). En general, este método arroja resultados satisfactorios para la mayoría de los modelos continuos y resulta ser muy bueno como primera aproximación cuando no se conoce mucho del sistema en estudio. No se recomienda para ecuaciones diferenciales rígidas.

Ejemplo 1. Obtener las raíces de los siguientes polinomios e interprete los resultados.

a) $x^5 - 9x^4 + 25x^3 - 5x^2 - 26x + 24 = 0$

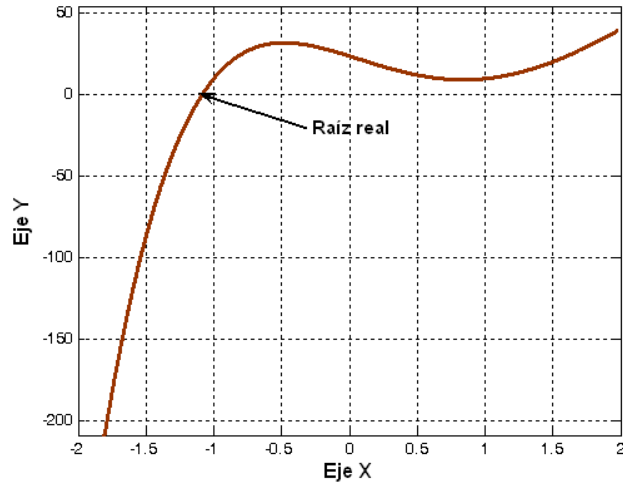


Fig. 1. Comportamiento de un polinomio de grado 5.

Gráficamente, en la Fig. 1 se observa que el polinomio tiene una raíz real y cuatro complejas, ya que sólo cruza el eje x una vez y las raíces complejas no cruzan el eje x. Las raíces obtenidas con el comando *roots* son:

```
p = [ 1 -9 25 -5 -26 24 ]; raices = roots(p)
raices =
4.1899 + 1.7423i    4.1899 - 1.7423i
-1.0883
0.8543 + 0.5842i    0.8543 - 0.5842i
```

b) $x^4 + 4x^3 + 6x^2 + 4x + 1 = 0$

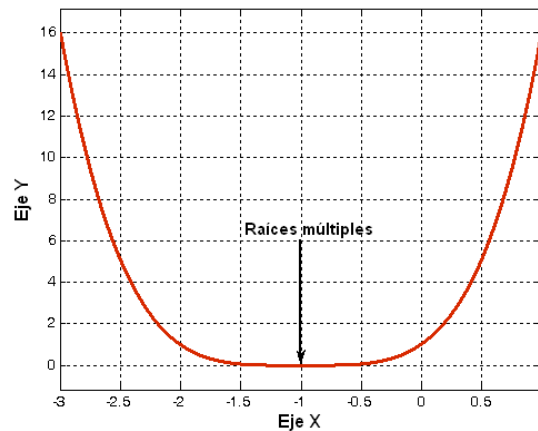


Fig. 2. Polinomio con cuatro raíces iguales (multiplicidad par).

En la Fig. 2 se observa que existen cuatro raíces múltiples (multiplicidad par), debido a que el polinomio toca en forma tangencial al eje x, pero no lo cruza. Calculando las raíces con el comando *roots* tenemos:

```
poli = [1 4 6 4 1]; raices = roots(poli)
raices =
-1 -1 -1 -1
```

c) $x^4 - 6x^3 + 12x^2 - 10x + 3 = 0$

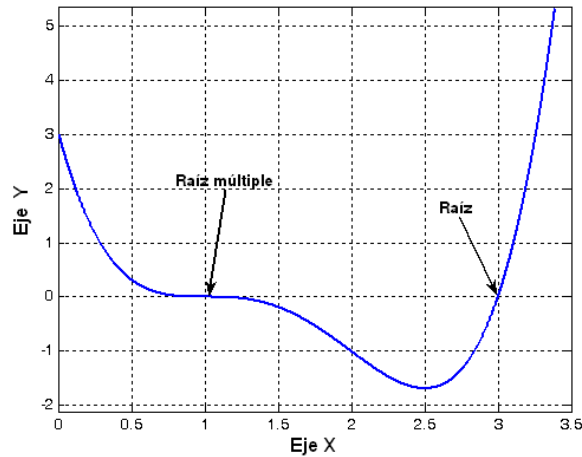


Fig. 3. Polinomio con tres raíces iguales (multiplicidad impar).

En la Fig. 3 se observa que existe una raíz en el entorno de tres y además existen tres raíces múltiples (multiplicidad impar) en el entorno de uno, debido a que la curva toca en forma tangencial al eje x y si lo cruza, a diferencia del caso anterior que no lo cruza. Las raíces obtenidas con el comando *roots* son:

```
poli = [1 -6 12 -10 3]; raices = roots(poli)
raices =
    3.0000 + 0.0000i
    1.0000 + 0.0000i
    1.0000 + 0.0000i
    1.0000 + 0.0000i
```

Ejemplo 2. A continuación se presentan los tres casos más comunes en la solución de sistemas de ecuaciones lineales.

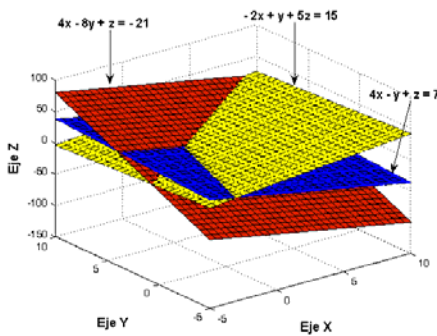


Fig. 4. Sistema de ecuaciones con solución única.

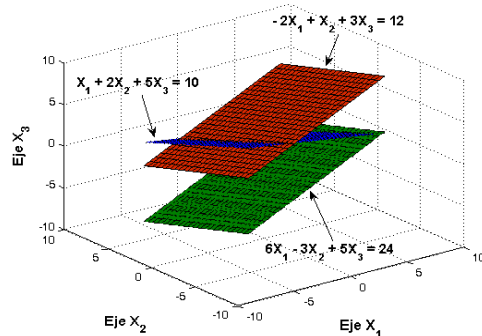


Fig. 5. Sistema de ecuaciones inconsistente.

Se puede observar en la Fig. 4 que los tres planos de las rectas se cruzan y por lo tanto existe un punto en común, es decir, el sistema es compatible determinado (solución única). En la Fig. 5 se observa, que los tres planos de las rectas nunca se cruzan y por lo tanto no existe un punto en común, es decir, el sistema es incompatible o inconsistente (no tiene solución). Con ayuda de MatLab y por medio del comando *rref* por el método de Gauss-Jordan tenemos:

$A = [4 \ -1 \ 1; 4 \ -8 \ 1; -2 \ 1 \ 5]; B = [7; -21; 15];$ $X = \text{rref}([A \ B])$ $X =$ $\begin{matrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 3 \end{matrix}$ La solución es $x = 2, y = 4, z = 3$	$A = [-2 \ 1 \ 3; 1 \ 2 \ 5; 6 \ -3 \ -9]; B = [12; 10; 24];$ $X = \text{rref}([A \ B])$ $X =$ $\begin{matrix} 1.0000 & 0 & -0.2000 & 0 \\ 0 & 1.0000 & 2.6000 & 0 \\ 0 & 0 & 0 & 1.0000 \end{matrix}$ Se observa en el tercer renglón que $0=1$, por lo que el sistema es incompatible, es decir, no tiene solución.
---	--

Se puede observar en la Fig. 6 que los tres planos son interceptados por una línea recta. Esto significa que el sistema tiene muchas soluciones, es decir, es compatible indeterminado y todas las soluciones se encuentran sobre la línea recta. Resolviendo el sistema por medio del comando *rref* de MatLab tenemos:

$A=[3 \ 2 \ -5; -1 \ 2 \ -1; -2 \ 1 \ 1]; B=[0; -8; -7]; X = \text{rref}([A \ B])$ $X =$ $\begin{matrix} 1 & 0 & -1 & 2 \\ 0 & 1 & -1 & -3 \\ 0 & 0 & 0 & 0 \end{matrix}$ Se observa en el tercer renglón que $0=0$, por lo que el sistema es indeterminado.
--

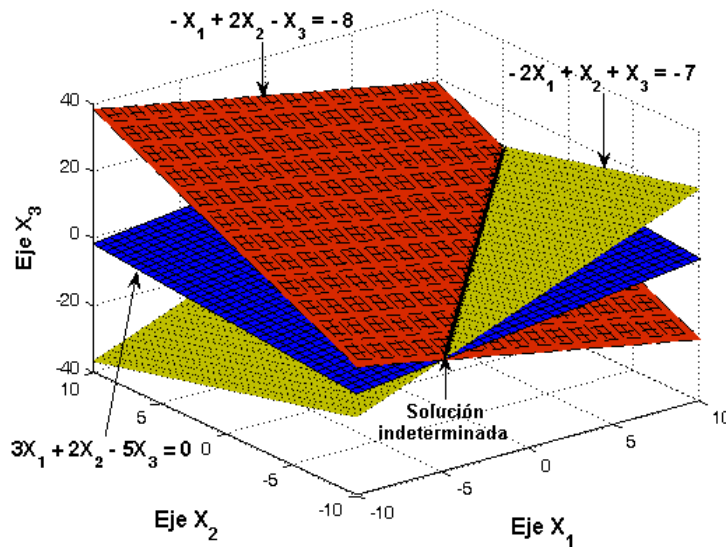


Fig. 6. Sistema de ecuaciones indeterminado.

Ejemplo 3. Resuelva el siguiente sistema de ecuaciones no lineales.

$$7x^3 - 10x - y = 1$$

$$8y^3 - 11y^2 + x = 1$$

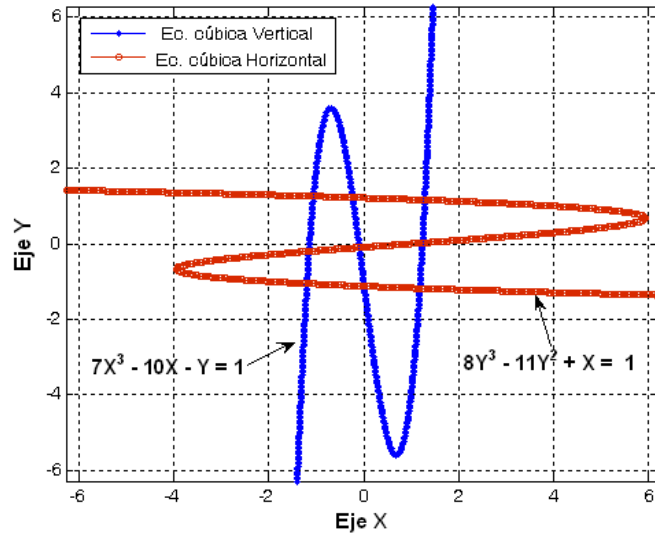


Fig.7. Solución gráfica de un sistema de dos polinomios cúbicos

La Fig. 7 muestra que hay nueve puntos donde se cruzan ambas gráficas. Se calculará la solución cercana a (1,0). Se genera un archivo en el editor de M-File de MatLab con el sistema de ecuaciones no lineales con el nombre de *cubicos.m* y se guarda en Work.

```
%Solución de un sistema de dos ecuaciones no lineales
%Polinomios cúbicos
function F = cubicos(x)
F = [7*(x(1)^3) - 10*x(1) - x(2)- 1;
     8*(x(2)^3) - 11*x(2) + x(1)- 1];
```

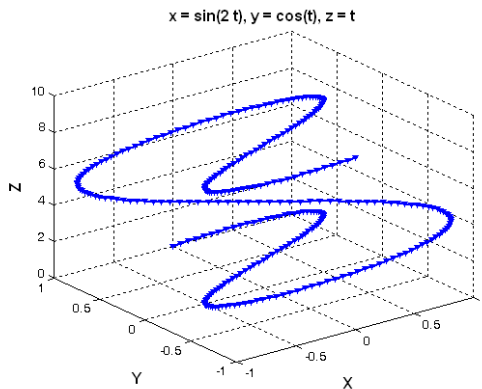
Escribiendo las instrucciones en la Ventana de Comandos de MatLab para llamar al archivo *cubicos.m* y resolver el sistema tenemos:

```
[x fval] = fsolve('cubicos',[1,0])
Optimization terminated: first-order optimality is less than options.TolFun.
x =
    1.2434    0.0221
fval =
    1.0e-012 *
    0.1565
    0
La solución es X = 1.2434 Y = 0.0221
```

Ejemplo 3. Casos de integración numérica.

a) Calcular la longitud de la curva espacial representada por las ecuaciones paramétricas $x(t) = \sin(2t)$, $y(t) = \cos(t)$, $z(t) = t$ donde $t \in [0, 3\pi]$. La representación gráfica se

muestra en la Fig. 7, así como su solución mediante el método de Lobatto con el comando *quadl*:



```
L= quadl(inline('sqrt(4*cos(2*t).^2 +
                sin(t).^2 + 1)'), 0,3*pi)
L =
    17.2220
```

Figura 8. Curva espacial

b) Obtener el valor de la siguiente integral doble

$$\int_{-\pi}^{3\pi/2} \int_0^{2\pi} [x \cos(y) + y \sin(x)] dy dx$$

Graficando $z = f(x, y) = x \cos y + y \sin x$, se tiene la Fig. 8 y su solución mediante doble integración numérica por el método de Simpson y el comando *dblquad* de MatLab:

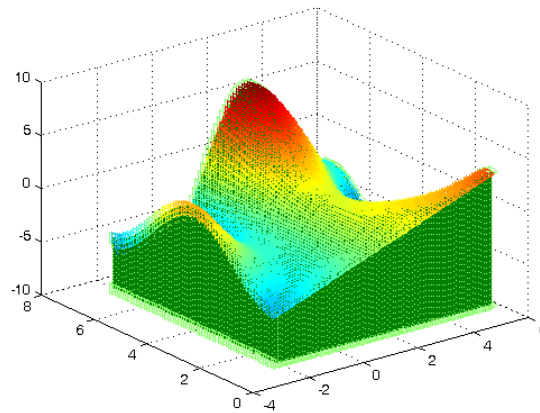


Figura 8. Área de superficie con contorno

```
int_doble= dblquad(inline('x*cos(y) + y*sin(x)'), -pi,3*pi/2,0,2*pi)
int_doble =
    -19.7392
```

c) Obtener el valor de la siguiente integral triple:

$$\int_{-2}^2 \int_{-2}^2 \int_{-2}^2 (x^2 + y^2 + z^2 - 1) dx dy dz$$

La representación gráfica de la integral triple se presenta mediante superficies de nivel y en forma de unidad como se muestra en la Fig. 9. La integral se debe interpretar como toda la materia que está dentro de la pelota.

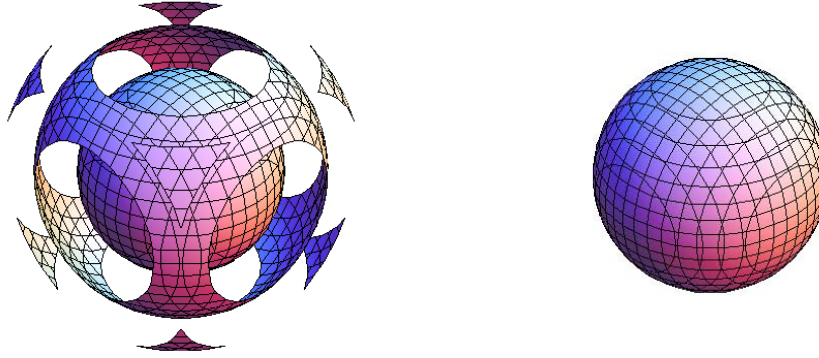


Fig. 9 Pelota mediante superficies de nivel y en forma sólida

```
pelota = tripequad(inline('x.^2+y.^2+z.^2-1'),-2, 2, -2, 2, -2, 2
pelota =
192
```

Ejemplo 4. La representación mediante variables de estado del sistema mecánico de la Fig. 10 es:

$$\frac{dx_1}{dt} = s_1; \quad \frac{ds_1}{dt} = \frac{k}{m_1}(x_2 - x_1); \quad \frac{dx_2}{dt} = s_2; \quad \frac{ds_2}{dt} = -\frac{B}{m_2}s_2 - \frac{k}{m_2}(x_2 - x_1) + \frac{f(t)}{m_2}$$

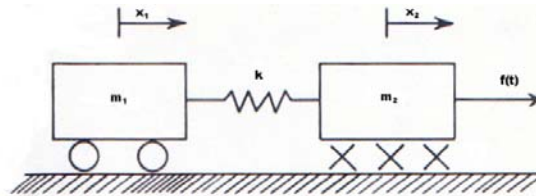


Fig. 10. Sistema mecánico

donde la masa m_i tiene el desplazamiento x_i y la velocidad s_i , k es la constante del resorte y B es el factor de amortiguamiento viscoso. Determinar la respuesta libre del sistema [$f(t) = 0$], para los siguientes valores: $M_1=0.9$; $M_2=0.1$; $K=0.9$; $B=1.5$; $t_0=0.0$; $t_f=10.0$ y los valores iniciales $X_1(t_0)=0$; $S_1(t_0)=1$; $X_2(t_0)=2$; $S_2(t_0)=1.5$. Considere como salidas del sistema las velocidades S_1 y S_2 y los desplazamientos X_1 y X_2 . Se genera un programa archivo para resolver el sistema de ecuaciones diferenciales dado y se guarda con el nombre de *sistema_mecanico.m*.

```
% Problema del sistema mecánico de dos masas
function dy = sistema_mecanico(t,y)
dy = zeros(4,1); %vector columna
dy(1) = y(3); dy(2) = y(4);
dy(3) = (y(2) - y(1)); dy(4) = -(1.5/0.1)*y(4) - (0.9/0.1)*(y(2) - y(1));
```

Se ejecuta el programa *sistema_mecanico.m* para encontrar la solución del sistema de ecuaciones diferenciales, por medio del Método de Runge-Kutta de 4º orden y el comando *ode45* de MatLab:

```
[T, Y] = ode45( @sistema_mecanico, [0 10], [0 2 1 1.5] )
```

Graficando las velocidades S_1 y S_2 y los desplazamientos X_1 y X_2 de las masas m_1 y m_2 se tienen la Fig. 11:

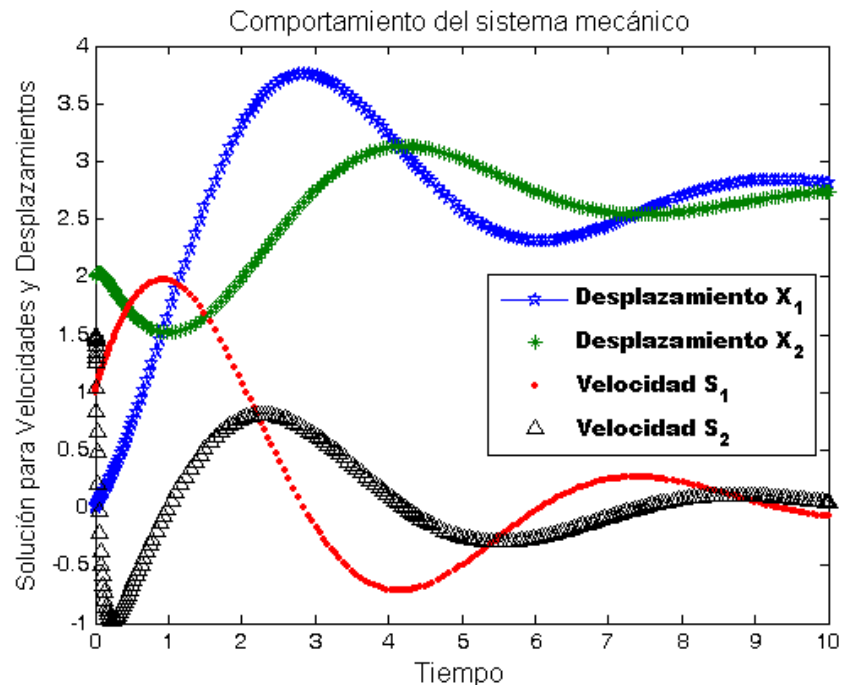


Fig.11. Comportamiento gráfico del sistema mecánico de dos masas

Se observa en la Fig. 11 que m_1 tiene un desplazamiento y una velocidad mayor cercana a los 3seg y a 1seg respectivamente y m_2 tiene un desplazamiento y una velocidad mayor cercana a los 4seg y a 2.5seg. Esto es, al principio m_1 se mueve más rápido que m_2 , aunque después para fines prácticos se estabilizan.

CONCLUSIONES.

El uso de las tecnologías y en especial el software MatLab, como un recurso didáctico, para la enseñanza-aprendizaje de los Métodos Numéricos, representa un verdadero reto para los académicos, ya que deben buscar estrategias que permitan su integración en el aprendizaje de los alumnos y esto requiere esfuerzo y dedicación. Se debe transformar el salón de clases en un laboratorio permitiendo que los alumnos reflexionen de manera autónoma y crítica sobre las actividades que realizan durante su aprendizaje y que además puedan comprobar los procesos realizados para llegar a la solución de un problema, mediante los Métodos Numéricos adecuados. MatLab empleado adecuadamente puede individualizar el proceso de enseñanza, facilitando la adaptación curricular a las necesidades e intereses de cada alumno, trasladando así soluciones y estrategias desde los contextos teóricos originales a otros nuevos mucho más inteligibles para él. El académico debe tener siempre presente de usar recursos didácticos que le permitan mejorar su quehacer, ya que las nuevas tecnologías son al mismo tiempo el resultado de la evolución de las matemáticas, y también el motor de esa evolución. Nuestra educación debe ser cambiante, evolucionar con la tecnología.

BIBLIOGRAFÍA

The MathWorks, Inc. *Matlab user's guide*. USA: Ed. Massachusetts, 2009.