

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA-LEÓN
FACULTAD DE CIENCIAS
DEPARTAMENTO DE ESTADÍSTICA-MATEMÁTICA



**SOLUCIONES NUMÉRICAS DE SISTEMAS DE ECUACIONES
DIFERENCIALES ORDINARIAS**

**MONOGRAFÍA
PARA OPTAR AL TÍTULO DE LICENCIADO EN
MATEMÁTICA**

PRESENTADO POR:

Bra: SILVIA ELENA REYES Z.
Br: FERNANDO JOSÉ PÉREZ P.
Br: MARTÍN JOSÉ ALONSO C.

TUTOR:
MSC. FELIPE CAMPOS ALVAREZ

Diciembre 2004

INDICE

Introducción	1
Objetivos	2
Primer Capitulo	
	3
Sistemas de ecuaciones diferenciales ordinarias lineales de primer orden	
Teorema número 1	5
Teorema número 2	6
Problemas de valores en la frontera	7
Solución del sistema lineales por eliminación	8
Ejemplo número 1	8
Ejemplo número 2	13
Segundo Capitulo	
Métodos numéricos para resolver sistemas de ecuaciones diferenciales lineales	17
Aspecto de análisis numéricos	17
Errores	18
	21
Método de Euler	
Método del Euler Introducción	21
Algoritmo del método de Euler para sistemas de ecuaciones diferenciales ordinarias	25
Programa del método de Euler	27
Notación (de la salida)	30
Tabla #1 Salida del programa de Euler	31

Método de Runge-Kutta	32
Introducción Método de Runge-Kutta	32
Algoritmo de Runge Kutta para sistema de ecuaciones diferenciales ordinarias de primer orden	34
Programa de Runge Kutta	36
Notación (salida)	39
Tabla #2 Salida del programa de Runge-Kutta	40
Método Predictor Corrector Adams-Bashforth y Adams-Moulton	41
(Adams-Bashforth y Adams-Moulton)	41
Algoritmo Predictor-Corrector de Adams-Basforth para sistema ecuaciones diferenciales ordinarias de primer orden	45
Programa Predictor-Corrector de Adams-Basforth	48
Notación (salida)	52
Tabla #3 Adams-Basforth -Moulton	53
Método Predictor -Corrector Milne-Simpson	54
Predictor-Corrector de Milne-Simpson	54
Algoritmo Predictor-Corrector Milne-Simpson para Sistemas de Ecuaciones Diferenciales de primer orden.	56
Programa Predictor-Corrector Milne-Simpson	59
Tabla #4 de Milne-Simpson	64
Conclusiones	66
Bibliografía	67

INTRODUCCION

Resolver ecuaciones diferenciales es uno de los problemas principales de los análisis numéricos. Esto ocurre porque hay una gran variedad de aplicaciones que al formularse conducen a ecuaciones diferenciales, pocas de las cuales pueden resolverse analíticamente.

Las soluciones de sistemas de dos ecuaciones diferenciales ordinarias se enfocan en como resolver un sistema de dos ecuaciones de primer orden. En los análisis numéricos se diseñan métodos para aproximar de manera eficientes las soluciones de las ecuaciones.

La eficiencia de estos métodos depende tanto de la precisión como de la facilidad con que se implementen para obtener la aproximación, se idea el método de algoritmo, esta es una secuencia de operaciones algebraicas y logística que producen la aproximación al sistema con precisión predeterminada.

En el presente trabajo daremos una breve descripción del método analítico de eliminación de sistemas de ecuaciones diferenciales ordinarias lineales y resolvemos algunos ejercicios sencillos con el fin de ilustrar tal método.

Seguidamente presentaremos una descripción breve de los algoritmos para los métodos numéricos clásicos para resolver sistemas de ecuaciones diferenciales ordinarias como, Método de Euler, Método de Runge-Kutta, Método Predictor Corrector de Adams Bashforh y Método Predictor Corrector de Milne-Simpson. Elaboramos programas en Pascal para tales métodos y daremos algunos ejemplos.

OBJETIVOS

OBJETIVOS GENERALES:

- Estudiar y dominar los algoritmos de los métodos numéricos clásicos para resolver sistemas de ecuaciones diferenciales ordinarias.

OBJETIVOS ESPECIFICOS:

- Elaborar los programas en Pascal de los métodos numéricos clásicos para resolver sistemas de ecuaciones diferenciales ordinarias.
- Comparar las ventajas de un método u otro en la resolución de un mismo problema en lo que se refiere a eficiencia y exactitud.

SISTEMAS DE ECUACIONES DIFERENCIALES ORDINARIAS LINEALES DE PRIMER ORDEN.

Existe una importante relación entre los sistemas de ecuaciones diferenciales ordinarias y las ecuaciones diferenciales aislada de orden arbitrario. En efecto de n -ésimo orden

$$y^{(n)} = F(t, y, y', \dots, y^{(n-1)}) \quad (1.1)$$

Siempre puede reducirse a un sistema de n ecuaciones de primer orden que tiene una forma un tanto especial. Para mostrar esto, introduciremos las variables x_1, x_2, \dots, x_n definidas por

$$x_1 = y, \quad x_2 = y', \quad x_3 = y'', \quad \dots, \quad x_n = y^{(n-1)} \quad (1.2)$$

Inmediatamente se deduce que

$$\begin{aligned} x_1' &= x_2, \\ x_2' &= x_3, \\ &\cdot \\ &\cdot \\ x_{n-1}' &= x_n, \end{aligned} \quad (1.3)$$

y de la ecuación (1.1) se obtiene

$$x_n' = F(t, x_1, x_2, \dots, x_n) \quad (1.4)$$

Las ecuaciones (1.3) y (1.4) son un caso especial del sistema más general (Forma Normal)

$$\begin{aligned}x'_1 &= F_1(t, x_1, \dots, x_n), \\x'_2 &= F_2(t, x_1, \dots, x_n), \\&\cdot \\&\cdot \\x'_n &= F_n(t, x_1, \dots, x_n)\end{aligned}\tag{1.5}$$

Ejemplo: Convierta el problema con valores iniciales

$$y''+3y'+2y = 0 ; \quad y(0) = 1; \quad y'(0) = 1$$

en un problema con valores iniciales para un sistema en forma normal.

Solución:

Primero expresamos la ecuación diferencial como

$$y'' = -3y' - 2y$$

Al hacer $x'_1 = y$ y $x_2 = y'$ obtenemos

$$\begin{aligned}x'_1 &= x_2 \\x'_2 &= -3x_2 - 2x_1\end{aligned}$$

Las y las condiciones iniciales se transforman en $x_1(0)=1, x_2(0)=1$.

Se dice que el sistema de ecuaciones (1.5) tiene una solución sobre el intervalo $\alpha < t < \beta$ si existe un conjunto de n funciones $x_1 = \phi_1(t), \dots, x_n = \phi_n(t)$, que sean diferenciales en todos los puntos en $\alpha < t < \beta$ y que satisfaga el sistema (1.5) en todos los puntos en este intervalo.

Además del sistema dado de ecuaciones diferenciales, también puede darse condiciones iniciales de la forma:

$$x_1(t_0) = x^0_1, x_2(t_0) = x^0_2, \dots, x_n(t_0) = x^0_n \tag{1.6}$$

Donde t_0 es un valor especificado de t en $\alpha < t < \beta$ y x_1^0, \dots, x_n^0 son números prescrito. Las ecuaciones diferenciales (1.5) y las condiciones iniciales (1.6), juntas, forman sistema de ecuaciones diferenciales con valores iniciales.

A continuación daremos un par de teoremas muy importante en el estudio de sistema de ecuaciones diferenciales ordinarias sin demostración

Teorema: 1

Suponga que cada una de las funciones F_1, \dots, F_n y cada una de las derivadas parciales $\partial F_1 / \partial x_1, \dots, \partial F_n / \partial x_n$ son continuas en una región R del espacio t, x_1, x_2, \dots, x_n que contiene al punto $(t_0, x_1^0, x_2^0, \dots, x_n^0)$. Entonces existe un intervalo $|t - t_0| < h$ en el que existe una solución única $x_1 = \phi_1(t), \dots, x_n = \phi_n(t)$ del sistema de ecuaciones diferenciales (1.5), la cual también satisface las condiciones iniciales (1.6).

Sin embargo, que en la hipótesis del teorema nada se dice acerca de las derivadas parciales de las funciones F_1, F_2, \dots, F_n con respecto a t .

También, en la conclusión, no se especifica exactamente la longitud $2h$ del intervalo en donde existe la solución y, en algunos casos, puede ser muy corto. Finalmente, el teorema no es el más general que se conoce y las condiciones dadas son suficiente, pero no necesarias, para asegurar la conclusión.

Si cada una de las funciones F_1, F_2, \dots, F_n en las ecuaciones (1.5) es una función lineal de las variables dependientes x_1, \dots, x_n , entonces se dice que el sistema de ecuaciones es lineal. Por lo tanto, el sistema más general de n ecuaciones lineales de primer orden tiene la forma: (*Sistema de la Forma Normal o Canónica*)

$$x'_1 = p_{11}(t)x_1 + \dots + p_{1n}(t)x_n + g_1(t)$$

$$\begin{aligned}x'_2 &= p_{21}(t)x_1 + \dots + p_{2n}(t)x_n + g_2(t) \\ &\vdots \\ &\vdots \\ x'_n &= p_{n1}(t)x_1 + \dots + p_{nn}(t)x_n + g_n(t).\end{aligned}\tag{1.7}$$

Si cada una de las funciones g_1, \dots, g_n es idénticamente cero, entonces se dice que el sistema (1.7) es homogéneo; de lo contrario es no homogéneo.

Para el sistema lineal (1.7), el teorema de existencia y unicidad es más sencillo y tiene una conclusión más fuerte.

Teorema 2

Si las funciones $p_{11}, \dots, p_{nn}, g_1, \dots, g_n$ son continuas sobre un intervalo abierto $\alpha < t < \beta$, que contiene al punto $t = t_0$, entonces existe una solución única $x_1 = \phi_1(t), \dots, x_n = \phi_n(t)$, del sistema de ecuaciones diferenciales (1.7), que también satisface las condiciones iniciales (1.6). Esta solución es validada en todo el intervalo $\alpha < t < \beta$.

En contraste con la situación que prevalece en un sistema no lineal, la existencia y la unicidad de la solución de un sistema lineal queda garantizada en todo el intervalo en el que se satisface la hipótesis. Además, para un sistema lineal, los valores iniciales x^0_1, \dots, x^0_n en $t = t_0$ son completamente arbitrario, mientras que en el caso no lineales, el punto inicial debe estar en la región R que se define en el teorema 1.

PROBLEMAS DE VALORES EN LA FRONTERA

Otro tipo de problema consiste en resolver una ecuación diferencial de orden dos o de orden mayor que dos en la cual la variable dependiente y (o sus derivadas) se especifica en dos *puntos diferentes*.

Un problema como:

$$\text{Resolver: } a_2(x) \frac{d^2y}{dx^2} + a_1(x) \frac{dy}{dx} + a_0(x)y = g(x)$$

$$\text{Sujeta a : } y(a) = y_0, y(b) = y_1$$

Se llaman problemas de valores de frontera de dos puntos o, simplemente, un *problema de valores de frontera*.

Ejemplo: En el caso del problema de valor de frontera.

$$x^2y'' - 2xy' + 2y = 6$$

$$y(1) = 0, y(2) = 3$$

Se busca una función definida en un intervalo que contenga $x = 1$ y $x = 2$, que satisfaga la ecuación diferencial y cuya gráfica pase por los dos puntos $(1,0)$ y $(2,3)$.

Los problemas de valor de frontera se encuentran a menudo en las aplicaciones de ecuaciones diferenciales parciales.

SOLUCION DE SISTEMAS LINEALES POR ELIMINACION.

La forma más elemental de resolver un sistema de ecuaciones diferenciales lineales, con coeficientes constante es el proceso directo de eliminación. Esto es, las variables dependientes que se eliminan sucesivamente con el objeto de obtener una sola ecuación de orden superior que solo contenga la variable dependiente.

Para sistema de dos ecuaciones de primer orden tales métodos son bastante eficientes pueden aplicarse tanto a sistema no homogéneo como a sistema homogéneo y también pueden usarse con algunas ventajas en problemas que comprendan sistema de ecuaciones diferenciales de orden superior, sin embargo los propios métodos de eliminación no llevan por si mismo a discusión teóricas, ni son convenientes para resolver grandes sistemas de ecuaciones diferenciales.

A continuación desarrollaremos un par de ejemplos para explicar el método de eliminación.

Ejemplo # 1. Resolver el siguiente sistema de ecuaciones sujeto a las condiciones dadas.

$$(1) \quad x' = -4x + 3y + 6$$

$$(2) \quad y' = -2.4x + 1.6y + 3.6$$

$$\text{Condiciones iniciales: } x(0) = 0$$

$$y(0) = 0$$

Solución:

Despejando “y “ en (1),

$$x' = -4x + 3y + 6;$$

Monografía para optar al título de Licenciatura en Matemática.

$$3y = x' + 4x - 6;$$

$$y = \frac{1}{3}x' + \frac{4}{3}x - 2 \quad (3);$$

Derivando (3);

$$y' = \frac{1}{3}x'' + \frac{4}{3}x' \quad (4);$$

Sustituyendo (3) y (4) en (2);

$$y' = -2.4x + 1.6y + 3.6$$

$$\frac{1}{3}x'' + \frac{4}{3}x' = -2.4x + 1.6\left(\frac{1}{3}x' + \frac{4}{3}x - 2\right) + 3.6$$

$$\frac{1}{3}x'' + \frac{4}{3}x' = -2.4x + \frac{1.6}{3}x' + \frac{6.4}{3}x - 3.2 + 3.6$$

Multiplicando la ecuación anterior por 3 obtenemos

$$x'' + 4x' = -7.2x + 1.6x' + 6.4x - 9.6 + 10.8 \quad \text{Reduciendo términos}$$

$$x'' + 4x' = 1.6x' - 0.8x + 1.2 \quad \text{Ordenando y reduciendo}$$

$$x'' + 2.4x' + 0.8x = 1.2 \quad (5)$$

Resolviendo la ecuación homogénea de (5)

$$x'' + 2.4x' + 0.8x = 0$$

Su ecuación característica es:

$$m^2 + 2.4m + 0.8 = 0 \quad \text{Factorizando}$$

$$(m + 2)(m + 0.4) = 0 \quad \text{Igualando a cero}$$

$$m_1 + 2 = 0, \quad m_2 + 0.4 = 0$$

$$m_1 = -2, \quad m_2 = -0.4$$

m_1 y m_2 son raíces reales y diferentes por tanto la solución es del tipo

$$x(t) = c_1 e^{m_1 t} + c_2 e^{m_2 t}, \quad \text{esto es,}$$

$$x(t) = c_1 e^{-2t} + c_2 e^{-0.4t} \quad (6)$$

La ecuación (6) es una solución complementaria de la ecuación (1)

Resolviendo la ecuación no homogénea (5)

$$x'' + 2.4x' + 0.8x = 1.2$$

Utilizando los operadores

$$D(D^2x + 2.4Dx + 0.8x) = D1.2$$

$$D(D^2 + 2.4D + 0.8)x = 0$$

$$D(D^2 + 2.4D + 0.8) = 0$$

La ecuación característica es:

$$m(m^2 + 2.4m + 0.8) = 0 \quad \text{Factorizando}$$

$$(m + 2)(m + 0.4)m = 0$$

$$m_1 + 2 = 0 \quad m_2 + 0.4 = 0 \quad m_3 = 0$$

$$m_1 = -2 \quad m_2 = -0.4$$

La solución es del tipo

$$x(t) = c_1 e^{m_1 t} + c_2 e^{m_2 t} + c_3 e^{m_3 t}$$

$$x(t) = c_1 e^{-2t} + c_2 e^{-0.4t} + c_3 e^{0t}$$

$$x_g(t) = c_1 e^{-2t} + c_2 e^{-0.4t} + c_3 \quad (7)$$

$$x_c = c_1 e^{-2t} + c_2 e^{-0.4t} \quad \text{solución complementaria}$$

$$x_p = c_3 \quad \text{solución particular}$$

Tomando la solución particular con su primera y segunda derivada,

$$x_p = c_3 \quad (8)$$

$$x'_p = 0 \quad (9)$$

$$x''_p = 0 \quad (10)$$

Sustituyendo las ecuaciones (8), (9) y (10) en la ecuación (5)

$$x''_p + 2.4x'_p + 0.8x_p = 1.2$$

$$0 + 2.4(0) + 0.8c_3 = 1.2$$

$$c_3 = \frac{1.2}{0.8}$$

$$c_3 = 1.5$$

Sustituyendo c_3 en (7) obtenemos

$$x_g(t) = c_1e^{-2t} + c_2e^{-0.4t} + 1.5 \quad (11)$$

Derivando la ecuación anterior

$$x'_g(t) = -2c_1e^{-2t} - 0.4c_2e^{-0.4t} \quad (12)$$

Sustituyendo las ecuaciones (11) y (12) en (1)

$$x' = -4x + 3y + 6 ;$$

$$-2c_1e^{-2t} - 0.4c_2e^{-0.4t} = -4[c_1e^{-2t} + c_2e^{-0.4t} + 1.5] + 3y + 6$$

Reduciendo términos semejantes y despejando " y "

$$-2c_1e^{-2t} - 0.4c_2e^{-0.4t} = -4c_1e^{-2t} - 4c_2e^{-0.4t} - 6 + 3y + 6$$

$$3y = 4c_1e^{-2t} + 4c_2e^{-0.4t} - 2c_1e^{-2t} - 0.4c_2e^{-0.4t}$$

$$3y = 2c_1e^{-2t} + 3.6c_2e^{-0.4t}$$

$$y_g(t) = \frac{2}{3}c_1e^{-2t} + \frac{3.6}{3}c_2e^{-0.4t} \quad (13)$$

Tomando las soluciones generales (11), (13) y sustituyendo las condiciones iniciales respectivamente $x(0)$, $y(0)$ obtenemos

$$x_g(0) = c_1e^{-2(0)} + c_2e^{-0.4(0)} + 1.5 \quad \text{haciendo } x_g(0) = 0$$

$$c_1 + c_2 + 1.5 = 0$$

$$c_1 + c_2 = -1.5 \quad (14)$$

$$y_g(0) = \frac{2}{3}c_1e^{-2(0)} + \frac{3.6}{3}c_2e^{-0.4(0)} = 0$$

$$\frac{2}{3}c_1 + \frac{3.6}{3}c_2 = 0 ; \text{ multiplicando por 3}$$

$$2c_1 + 3.6c_2 = 0 \quad (15)$$

Resolviendo las ecuaciones (14) y (15)

$$c_1 + c_2 = -1.5 \quad (14)$$

$$2c_1 + 3.6c_2 = 0 \quad (15)$$

La ecuación (14) la multiplicamos por -2

$$-2c_1 - 2c_2 = 3$$

$2c_1 + 3.6c_2 = 0$; Eliminando c_1 nos resulta

$$1.6c_2 = 3 \quad \text{Despejando } c_2$$

$$c_2 = \frac{3}{1.6}$$

$$c_2 = 1.875$$

Sustituyendo c_2 en (14)

$$c_1 + c_2 = -1.5$$

$$c_1 + 1.875 = -1.5$$

$$c_1 = -1.5 - 1.875$$

$$c_1 = -3.375$$

Sustituyendo c_1 y c_2 en (11)

$$x(t) = c_1 e^{-2t} + c_2 e^{-0.4t} + 1.5$$

$$x(t) = -3.375 e^{-2t} + 1.875 e^{-0.4t} + 1.5$$

Sustituyendo c_1 y c_2 en (13)

$$y(t) = \frac{2}{3} c_1 e^{-2t} + 1.2 c_2 e^{-0.4t}$$

$$y(t) = \frac{2}{3} (-3.375) e^{-2t} + 1.2(1.875) e^{-0.4t}$$

$$y(t) = -2.25 e^{-2t} + 2.25 e^{-0.4t}$$

Las soluciones son:

$$x(t) = -3.375 e^{-2t} + 1.875 e^{-0.4t} + 1.5$$

$$y(t) = -2.25 e^{-2t} + 2.25 e^{-0.4t}$$

Ejemplo # 2. Resolver el siguiente sistema de ecuaciones sujeto a las condiciones dadas.

$$(1) \quad \mathbf{x}' = \mathbf{x} + \mathbf{y} \quad \mathbf{x}(0) = \mathbf{1}$$

$$(2) \quad \mathbf{y}' = 3\mathbf{x} - \mathbf{y} \quad \mathbf{y}(0) = \mathbf{0}$$

Tomamos la ecuación (1) y despejamos " y "

$$x' = x + y$$

$$y = x' - x \quad (3) \quad \text{derivando 'y'}$$

$$y' = x'' - x' \quad (4)$$

Luego sustituyendo en la ecuación (2) las ecuaciones (3) y (4)

$$y' = 3x - y$$

$$x'' - x' = 3x - (x' - x)$$

$$x'' - x' = 3x - x' + x \quad \text{reduciendo términos semejantes}$$

$$x'' = 4x$$

$$x'' - 4x = 0 \quad (5)$$

Su ecuación característica es:

$$m^2 - 4 = 0; \quad \text{factorizando};$$

$$(m - 2)(m + 2) = 0; \quad \text{Igualando ambos factores a cero};$$

$$m - 2 = 0;$$

$$m + 2 = 0;$$

$$m_1 = 2 \quad \text{y} \quad m_2 = -2;$$

Obtenemos raíces reales y diferentes por tanto la solución es del tipo:

$$x = c_1 e^{m_1 t} + c_2 e^{m_2 t}$$

$$x(t) = c_1 e^{2t} + c_2 e^{-2t} \quad (6)$$

es la solución general para x; derivando

$$x'(t) = 2c_1 e^{2t} - 2c_2 e^{-2t} \quad (7)$$

La ecuación (6) y (7) la sustituimos en la ecuación (3);

$$y = x' - x$$

$$y = 2c_1 e^{2t} - 2c_2 e^{-2t} - (c_1 e^{2t} + c_2 e^{-2t})$$

$$y = 2c_1 e^{2t} - 2c_2 e^{-2t} - c_1 e^{2t} - c_2 e^{-2t}$$

$$y = c_1 e^{2t} - 3c_2 e^{-2t}; \quad (8) \text{ solución general para 'y'}$$

Tomando las soluciones generales y empleando las condiciones iniciales para la ecuación (6):

$$x(0) = c_1 e^{2(0)} + c_2 e^{-2(0)}$$

$$x(t) = c_1 e^{2t} + c_2 e^{-2t}, \quad \text{tomando } x(0) = 1$$

$$x(0) = c_1 + c_2 = 1$$

$$c_1 + c_2 = 1 \quad (9)$$

$$\text{y para la ecuación (8) } y(t) = c_1 e^{2t} - 3c_2 e^{-2t}$$

$$y(0) = c_1 e^{2(0)} - 3c_2 e^{-2(0)}$$

$$c_1 - 3c_2 = 0 \quad (10)$$

De la ecuación (9) y (10) tenemos

$$c_1 + c_2 = 1$$

$$c_1 - 3c_2 = 0 \text{ la ecuación (9) la multiplicamos por } (-1)$$

$$-c_1 - c_2 = -1$$

$$c_1 - 3c_2 = 0 \text{ eliminamos } c_1 \text{ y nos queda}$$

$$-4c_2 = -1 \text{ despejamos } c_2; \quad c_2 = \frac{1}{4}$$

Ahora sustituimos c_2 en la ecuación (9)

$$c_1 + c_2 = 1$$

$$c_1 + \frac{1}{4} = 1$$

$$c_1 = 1 - \frac{1}{4}$$

$$c_1 = \frac{3}{4}$$

Monografía para optar al título de Licenciatura en Matemática.

Luego sustituimos las constantes ya encontradas en las soluciones generales, para obtener:

$$\mathbf{x(t)} = \frac{3}{4} \mathbf{e}^{2t} + \frac{1}{4} \mathbf{e}^{-2t}$$

soluciones analíticas

$$\mathbf{y(t)} = \frac{3}{4} \mathbf{e}^{2t} - \frac{3}{4} \mathbf{e}^{-2t}$$

Con estos ejemplos hemos tratado de explicar el funcionamiento del método de eliminación. En el capítulo que sigue nos ocuparemos de cuatro métodos numéricos para tales sistemas.

METODOS NUMERICOS PARA RESOLVER SISTEMAS DE ECUACIONES DIFERENCIALES LINEALES.

ASPECTOS DE ANÁLISIS NUMERICO

El análisis numérico trata de diseñar métodos para aproximar de una manera eficiente, las soluciones de problemas expresados matemáticamente. Para obtener tal aproximación se idea un método llamado algoritmo.

Un algoritmo es un procedimiento que describe, sin ninguna ambigüedad, una sucesión finita de pasos a realizar en un orden específico.

El objetivo del algoritmo será generalmente el de implementar un procedimiento numérico para resolver un problema o aproximar una solución del problema.

En los algoritmos que diseñamos no se utilizara una tolerancia en el proceso iterativo, sino se introducirá el número de pasos (N) deseados por el usuario.

Generalmente se seleccionan métodos que produzcan resultados confiables en cuanto a la propagación del error que puede ocurrir cuando se aplica métodos numéricos. Con ellos se tiene la oportunidad de aproximar con cierta precisión de algunos problemas que no pueden resolverse exactamente.

ERRORES

El análisis del error en resultados numéricos es fundamental para cualquier problema ya sea realizado con una calculadora o con ayuda de una computadora. Los datos de entrada rara vez son exactos, ya que a menudo se basan en experimentos o son estimados, y los procesos numéricos a su vez introducen errores de varios tipos.

TIPOS DE ERRORES

Existen tres tipos básicos de errores en una computación numérica: inherente, por truncamiento, y por redondeo.

Errores inherentes: Son errores que existen en los valores de los datos, causados por incertidumbre en las mediciones, por verdaderas equivocaciones, o por la naturaleza necesariamente aproximada de la representación, mediante un número finito de dígitos, de cantidades que no pueden representarse exactamente con los números de dígitos permisible. Ejemplo: una medición física, una distancia, un periodo de tiempo, no puede ser exacta.

Errores por truncamiento: Los términos omitidos introducen un error en los resultados calculados. Este error se llama error de truncamiento.

Errores por redondeo: Aun si suponemos que los datos de entrada no tienen errores inherente y aplicamos procesos de computación que son finitos y no tienen, por tanto, error de truncamiento, podemos introducir otras clases de errores al efectuar aritmética simple.

Cada uno de los tipos de errores anteriores se puede expresar en forma absoluta o en forma relativa.

I. Errores absolutos: El error absoluto en una cantidad es la diferencia entre el valor verdadero, suponiendo que se conoce, y una aproximación al valor verdadero. La notación ordinaria consiste en indicar el valor aproximado mediante una barra sobre el símbolo de la cantidad, y el error se indica mediante la letra e con un subíndice. Entonces, si x es el valor verdadero escribimos

$$x = \bar{x} + e_x$$

En esta expresión e_x es el error absoluto, que, repetimos se define como la diferencia entre el valor real y la aproximación:

$$e_x = x - \bar{x}$$

II. Errores relativos: El error relativo es el cociente del error absoluto entre la aproximación:

$$\text{error relativo} = \frac{e_x}{\bar{x}}$$

Parecería más razonable definirlo como error absoluto dividido entre el valor verdadero, pero generalmente no conocemos este. Todo lo que tenemos generalmente es un valor aproximado y una estimación del error o un límite al tamaño máximo del error. Si el error es pequeño, la diferencia en la definición no tiene influencia muy grande en el valor numérico del error relativo.

Monografía para optar al título de Licenciatura en Matemática.

El error absoluto y el error relativo son aproximadamente iguales para números cercanos a 1. Para números no cercanos a 1 puede haber una gran diferencia.

Antes de presentar los cuatro métodos que estudiaremos, es necesario hacer algunas observaciones concernientes a los sistemas de ecuaciones diferenciales de primer orden.

Un sistema de orden m de problemas de valor inicial de primer orden se puede expresar en la forma:

$$\begin{aligned}\frac{du_1}{dt} &= f_1(t, u_1, u_2, \dots, u_m) \\ \frac{du_2}{dt} &= f_2(t, u_1, u_2, \dots, u_m) \\ &\cdot \\ &\cdot \\ &\cdot \\ \frac{du_m}{dt} &= f_m(t, u_1, u_2, \dots, u_m)\end{aligned}\tag{2.1}$$

Para $a \leq t \leq b$, con las condiciones iniciales

$$\begin{aligned}u_1(a) &= \alpha_1 \\ u_2(a) &= \alpha_2 \\ &\cdot \\ &\cdot \\ u_m(a) &= \alpha_m\end{aligned}\tag{2.2}$$

Se pretende encontrar m funciones u_1, u_2, \dots, u_m que satisfagan las Ecuaciones diferenciales del sistema y las condiciones iniciales.

MÉTODO DE EULER.

Introducción Método de Euler

El método de Euler se usa poco en la práctica, la sencillez en la forma de obtenerlo puede servir para ilustrar las técnicas en la elaboración de algunos de los métodos mas avanzados, sin lo incomodo del álgebra que acompaña a esas elaboraciones.

El objetivo del método es obtener una aproximación al problema de valor inicial bien planteado.

$$\frac{du_1}{dt} = f_1(t, u_1, u_2, \dots, u_m), \quad u_1(a) = \alpha_1$$

$$\frac{du_2}{dt} = f_2(t, u_1, u_2, \dots, u_m), \quad u_2(a) = \alpha_2$$

.

.

.

$$\frac{du_m}{dt} = f_m(t, u_1, u_2, \dots, u_m), \quad u_m(a) = \alpha_m$$

en $a \leq t \leq b$.

No se obtendrá una aproximación continua a la solución de $u_1(t), u_2(t), \dots, u_m(t)$; se obtendrán aproximaciones de las u_j , $j = 1, 2, \dots, m$ en diversos valores llamados puntos de mallas en el intervalo $[a, b]$.

Una vez obtenida la aproximación a la solución en esos puntos, la solución aproximada en otros puntos del intervalo puede obtener por interpolación.

En este capítulo se hace la suposición de que los puntos de mallas están distribuidos uniformemente a lo largo del intervalo $[a, b]$. Esta condición se puede garantizar escogiendo un entero positivo N y seleccionando los puntos de mallas.

$(t_0, t_1, t_2, \dots, t_n)$ donde $t_i = a + i * h$ para cada $i = 0, 1, 2, \dots, N$.

La distancia común entre los puntos $h = (b - a)/N$, se llama tamaño de paso.

Usaremos el teorema de Taylor para obtener el método de Euler. Supongamos que las $u_1(t), u_2(t), \dots, u_m(t)$ son las únicas soluciones de (2.1) y tienen cada una dos derivadas continuas en $[a, b]$, de manera que para cada

$i = 0, 1, 2, \dots, N-1$.

$$u_1(t_{i+1}) = u_1(t_i) + (t_{i+1}-t_i)u'_1(t_i) + ((t_{i+1}- t_i)^2/2)u''_1(\xi_i).$$

$$u_2(t_{i+1}) = u_2(t_i) + (t_{i+1} - t_i) u'_2(t_i) + ((t_{i+1} - t_i)^2/2)u''_2(\xi_i)$$

.
.
.

$$u_m(t_{i+1}) = u_m(t_i) + (t_{i+1} - t_i)u'_m + ((t_{i+1} - t_i)^2/2)u''_m(\xi_i)$$

para un número ξ_i en (t_i, t_{i+1}) . Si $h = t_{i+1} - t_i$ entonces

$$u_1(t_{i+1}) = u_1(t_i) + h * u'_1(t_i) + (h^2/2)u''_1(\xi_i)$$

$$u_2(t_{i+1}) = u_2(t_i) + h * u'_2(t_i) + (h^2/2)u''_2(\xi_i)$$

.
.
.

$$u_m(t_{i+1}) = u_m(t_i) + h * u'_m(t_i) + (h^2/2)u''_m(\xi_i)$$

y puesto que las $u_1(t), u_2(t), \dots, u_m(t)$ satisfacen la ecuación (2.1)

$$\begin{aligned}
 u_1(t_{i+1}) &= u_1(t_i) + h * f_1(t_i, u_1(t_i)) + (h^2/2)u''_1(t_i) \\
 u_2(t_{i+1}) &= u_2(t_i) + h * f_2(t_i, u_2(t_i)) + (h^2/2)u''_2(\xi) \\
 &\cdot \\
 &\cdot \\
 &\cdot \\
 u_m(t_{i+1}) &= u_m(t_i) + h * f_m(t_i, u_m(t_i)) + (h^2/2)u''_m(\xi_i)
 \end{aligned}
 \tag{2.3}$$

El método de Euler elabora, $w_{1,i} \cong u_1(t_i)$, $w_{2,i} \cong u_2(t_i)$, \dots , $w_{m,i} \cong u_m(t_i)$

Para cada $i = 1, 2, \dots, N$, eliminando el término restante.

Así $w_{0,1} = \alpha_1$, $w_{0,2} = \alpha_2, \dots, w_{0,m} = \alpha_m$.

$$\begin{aligned}
 w_{i+1,1} &= w_{i,1} + h * f_1(t, w_{i,1}, w_{i,2}, \dots, w_{i,m}) \\
 w_{i+1,2} &= w_{i,2} + h * f_2(t, w_{i,1}, w_{i,2}, \dots, w_{i,m})
 \end{aligned}
 \tag{2.4}$$

$$\cdot \\
 \cdot \\
 \cdot \\
 w_{i+1,m} = w_{i,m} + h * f_m(t, w_{i,1}, w_{i,2}, \dots, w_{i,m})$$

para cada $i = 1, 2, \dots, N-1$.

La ecuación (2.4) se llama ecuación en diferencia asociada al método de Euler.

A continuación presentaremos el algoritmo del método de Euler para sistemas de dos Ecuaciones Lineales y posteriormente el programa en Pascal elaborado.

Monografía para optar al título de Licenciatura en Matemática.

Consideramos en nuestros algoritmos y programas sistemas de ecuaciones diferenciales de la forma:

$$u'_1 = f_1(t, x, y) \quad (2.5)$$

$$u'_2 = f_2(t, x, y)$$

Donde las tildes denotan las derivadas con respecto a t .

Supóngase que también se nos han dado las condiciones

$$u_1(t_0) = x(t_0) = x_0, \quad u_2(t_0) = y(t_0) = y_0 \quad (\text{donde } t_0 = a, \quad x_0 = \alpha_1, \quad y_0 = \alpha_2).$$

Si f_1 y f_2 y sus derivadas parciales son continuas todas en una región que contenga al punto (t_0, x_0, y_0) entonces los teoremas 1 y 2 del capítulo anterior garantizan la existencia y unicidad de las soluciones

$x(t) = u_1(t)$ y $y(t) = u_2(t)$ en una vecindad de t_0 sobre el eje de la t . Con esta seguridad, podemos proceder a analizar de la aproximación numéricas de la solución es $u_1(t)$ y $u_2(t)$ de un problema de valores iniciales para el sistema (2.5).

Algoritmo Del método de Euler para Sistema de Ecuaciones Diferenciales Ordinarias.

Para aproximar la solución de un sistema de m-ésimo orden de problemas de valores iniciales de primer orden.

$$u'_j = f(t, u_1, u_2, \dots, u_m); \quad j = 1, 2, \dots, m$$

$$a \leq t \leq b \quad u_j(a) = \alpha_j, \quad j = 1, 2, \dots, m$$

en $(N + 1)$ puntos igualmente espaciados en el intervalo $[a, b]$

ENTRADA; puntos extremos a, b ; número de ecuaciones m ; número de iteraciones N , condiciones iniciales $\alpha_1, \alpha_2, \dots, \alpha_m$.

SALIDA: aproximaciones w_j , a $u_j(t)$ en los $(n + 1)$ valores de t ;

Paso 1: Sea $h = (b - a)/N$;

$$t_0 = a,$$

Paso 2: $j = 1, 2, \dots, m$,

Sea $w_j = \alpha_j$

Paso 3: SALIDA $(t, w_1, w_2, \dots, w_m)$;

Monografía para optar al título de Licenciatura en Matemática.

Paso 4: Para $i = 1, 2, \dots, N$ realizar pasos 5-7

Pasos 5: para $j = 1, 2, \dots, m$ sea

$$w_j = w_j + h * f_j(t, w_1, w_2, \dots, w_m) \quad (\text{Calcula los } w_j)$$

Paso 6: Sea $t = a + i * h$; (Calcula t_i)

Paso 7: SALIDA (t, w_1, w_2, \dots, w_m)

Paso 8: PARAR.

```
PROGRAM SISTEMA_DE_ECUACIONES_DIFERENCIALES_EULER;  
(*PROGRAMA PARA RESOLVER SISTEMA DE ECUACIONES  
DIFERENCIALES*)  
(* x ' = f(t, x, y) *)  
(* y ' = g(t, x, y) *)  
(* a ≤ t ≤ b, x(a) = alfa1, x(b) = alfa2*);  
USES CRT;  
CONST MAX = 1200;  
VAR  
m, N, r, j, i: INTEGER;  
a, b, h, C1, C2: REAL;  
t, x, y, w1, w2: ARRAY [0..MAX]OF REAL;  
FUNCTION F (m: integer; t, x, y: REAL): REAL;  
BEGIN  
CASE m OF  
1:F: = - 4*x + 3*y + 6;  
2:F: = 2*x + 2*y;  
3:F: = x - 4*y;  
END;  
END;  
FUNCTION G(r:INTEGER; t,x,y: REAL):REAL;  
BEGIN  
CASE r OF  
1:G: = - 2.4* x + 1.6*y + 3.6;  
2:G: = 3*x + y;  
3:G: = x + y;  
END;
```

```
END;
BEGIN
CLRSCR;
WRITELN ('ELIJA UNA FUNCION...');
WRITELN ('1:F(x, y, t) = - 4*x + 3*y + 6 ');
WRITELN ('2:F(x, y, t) = 2*x + 2*y ');
WRITELN ('3:F(x, y, t) = x - 4*y ');
WRITELN;
WRITELN ('ESCOJA EL NUMERO DE LA FUNCION...');
READLN (m);
CLRSCR;
WRITELN ('ELIJA UNA FUNCION...');
WRITELN ('1:G(x, y, t) = - 2.4*x + 1.6*y + 3.6');
WRITELN ('2:G(x, y, t) = 3* x + y ');
WRITELN ('3:G(x, y, t) = x + y ');
WRITELN ('ESCOJA EL NUMERO DE LA FUNCION...');
READLN (r);
CLRSCR;

WRITELN ('INTRODUZCA EL INTERVALO [a, b]');

WRITE ('INTRODUZCA EL VALOR DE "a" = ');
READLN (a);
WRITE ('INTRODUZCA EL VALOR DE "b" = ');
READLN (b);

WRITELN ('INTRODUZCA LAS CONDICIONES INICIALES "alfa1" y "alfa2"
');
```

```
WRITE ('INTRODUZCA EL VALOR INICIAL DE "alfa1" = ');
READLN (C1);
WRITE ('INTRODUZCA EL VALOR INICIAL DE "alfa2" = ');
READLN (C2);
WRITE ('INTRODUZCA LA CANTIDAD DE PASOS "n" = ');
READLN (N);
h: = (b - a)/N;
t [0]: = a;
w1 [0]: = C1;
w 2 [0]: = C2;
WRITELN('i   t(i)   x (i)   y(i) ');
WRITELN(0, '   ',t[0]:2:4,'   ',w1 [0]:4:8,'   ',w2 [0]:4:8);
FOR i: = 1 TO N DO
BEGIN
w[i]: = w1 [i-1] + h*F(m,t[i-1], w1 [i-1], w2 [i-1]);
w2 [i]: = w2 [i-1] +h*G(r,t[i-1], w1 [i-1], w2 [i-1]);
t[i]: = a + i*h;
{w1 [i]: = w1 [i-1];
w2 [i]: = w 2 [i-1];}
WRITELN (i, ' ',t[i]:10:4,'   ',w1 [i]:4:6,'   ',w2 [i]:4:6);
END;
REPEAT UNTIL KEYPRESSED;
END.
```


NOTACIÓN:

Salida del programa de Euler

Se elegirán las funciones 1:f: = - 4*x + 3* y + 6;

$$2:f: = -2.4*x + 1.6 y + 3.6;$$

para formar así el sistema de ecuaciones diferenciales lineales, con las condiciones iniciales $X(0) = 0$; $Y(0) = 0$. en el intervalo $0 \leq t \leq 0.5$; ($a=0$ y $b=0.5$) tomando el número de pasos $N=10$ y el tamaño de paso será $h = 0.05$.

w1: valor aproximado de la 1ra. Ecuación diferencial

w2: valor aproximado de la 2da. Ecuación diferencial

x1: valor exacto de la 1ra. Ecuación diferencial

y1: valor exacto de la 2da. Ecuación diferencial

$|w1 - x1|$: error absoluto del valor aproximado y valor exacto de la 1ra. Ecuación.

$|w2 - y1|$: error absoluto del valor aproximado y valor exacto de la 2ª. Ecuación.

Solución exacta de la 1ra. Ecuación: $x(t) = -3.375e^{-2t} + 1.875e^{-0.4t} + 1.5$

Solución exacta de la 2da. Ecuación: $y(t) = -2.25e^{-2t} + 2.25e^{-0.4t}$

A continuación presentamos una captura de la salida del programa de Euler en Pascal

```
" METODO DE EULER "  
Introduzca el intervalo [a, b]  
INTRODUZCA EL VALOR DE "a"= 0  
INTRODUZCA EL VALOR DE "b"= 0.5  
INTRODUZCA LAS CONDICIONES INICIALES  
INTRODUZCA EL VALOR INICIAL DE "alfa1"= 0  
INTRODUZCA EL VALOR INICIAL DE "alfa2"= 0  
INTRODUZCA LA CANTIDAD DE PASOS "N"= 10  
  
i      t<i>      x<i>      y<i>  
0      0.0000    0.00000000  0.00000000  
1      0.0500    0.30000000  0.18000000  
2      0.1000    0.56700000  0.33840000  
3      0.1500    0.80436000  0.47743200  
4      0.2000    1.01510280  0.59910336  
5      0.2500    1.20194774  0.70521929  
6      0.3000    1.36734109  0.79740311  
7      0.3500    1.51348334  0.87711442  
8      0.4000    1.64235383  0.94566558  
9      0.4500    1.75573290  1.00423636  
10     0.5000    1.85522178  1.05388733
```

TABLA #1 MÉTODO DE EULER

t(i)	W1(i)	w2(i)	X1(i)	Y1(i)	w1 - x1	w2 - y2
0.00	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
0.05	0.30000000	0.18000000	0.28404623	0.16956282	$1.59 \cdot 10^{-2}$	$1.04 \cdot 10^{-2}$
0.10	0.56700000	0.33840000	0.53826391	0.31963204	$2.87 \cdot 10^{-2}$	$1.87 \cdot 10^{-2}$
0.15	0.80436000	0.47743200	0.76554701	0.45212920	$3.88 \cdot 10^{-2}$	$2.53 \cdot 10^{-2}$
0.20	0.01510280	0.59910336	0.96851299	0.56879168	$9.53 \cdot 10^{-1}$	$3.03 \cdot 10^{-2}$
0.25	1.20194774	0.70521929	1.14952918	0.67119021	$5.24 \cdot 10^{-2}$	$3.40 \cdot 10^{-2}$
0.30	1.36734109	0.79740311	1.31073655	0.76074480	$5.66 \cdot 10^{-2}$	$3.66 \cdot 10^{-2}$
0.35	1.51348334	0.87711442	1.45407129	0.83873910	$5.94 \cdot 10^{-2}$	$3.93 \cdot 10^{-2}$
0.40	1.64435383	0.94566558	1.58128435	0.90633336	$6.78 \cdot 10^{-2}$	$3.93 \cdot 10^{-2}$
0.45	1.75573290	1.00423636	1.69395904	0.96457624	$6.17 \cdot 10^{-2}$	$3.96 \cdot 10^{-2}$
0.50	1.85522178	1.05388733	1.79352705	1.01441545	$6.17 \cdot 10^{-2}$	$3.94 \cdot 10^{-2}$

METODO DE RUNGE-KUTTA

INTRODUCCIÓN DE RUNGE-KUTTA

Este método para resolver sistemas de ecuaciones diferenciales de primer orden es una simple generalización del método para una sola ecuación de primer orden.

Por ejemplo, el método clásico de Runge- Kutta de cuarto orden dado por

$$w_0 = \alpha$$

$$k_1 = h*f(t_i, w_i)$$

$$k_2 = h*f(t_i + h/2, w_i + k_1/2)$$

$$k_3 = h*f(t_i + h/2, w_i + k_2/2)$$

$$k_4 = h*f(t_i, w_i + k_3) \quad y$$

$$w_{i+1} = w_i + [k_1 + 2 k_2 + 2 k_3 + k_4] /6$$

Para cada $i = 0, 1, 2, \dots, N-1$ y empleado para resolver el problema de valor inicial de primer orden.

$$y' = f(t, y), \quad a \leq t \leq b, \quad y(a) = \alpha,$$

Puede generalizarse como sigue. Se elige un entero $N > 0$ y se hace $h = (b - a)/N$. Realizando una partición del intervalo $[a, b]$ en N sub-intervalos con los puntos mallas

$$t_j = a + j*h, \quad \text{para cada } j = 0, 1, 2, \dots, N.$$

Usando la notación w_{ij} para representar una aproximación a $u_i(t_j)$ para cada $j = 0, 1, 2, \dots, m$; esto es w_{ij} aproximara la i -ésima solución $u_i(t)$ de (2.1) en el j -ésimo punto de malla t_j . Para las condiciones iniciales hagamos

$$(2.6) \quad w_{1,0} = \alpha_1, w_{2,0} = \alpha_2, \dots, w_{m,0} = \alpha_m.$$

Si suponemos que los valores de $w_{i,j+1}, w_{2,j+1}, \dots, w_{m,j+1}$ se han calculado

Obtenemos $w_{i,j+1}, w_{2,j+1}, \dots, w_{m,j+1}$ calculando primero

$$(2.7) \quad k_{1,i} = h \cdot f_i(t_j, w_{1,j}, w_{2,j}, \dots, w_{m,j}) \text{ para cada } i = 1, 2, \dots, m.$$

$$(2.8) \quad k_{2,i} = h \cdot f_i(t_j + h/2, w_{1,j} + (k_{1,1})/2, w_{2,j} + (k_{1,2})/2, \dots, w_{m,j} + (k_{1,m})/2)$$

para cada $i = 1, 2, \dots, m$.

$$(2.9) \quad k_{3,i} = h \cdot f_i(t_j + h/2, w_{1,j} + (k_{2,1})/2, w_{2,j} + (k_{2,2})/2, \dots, w_{m,j} + (k_{2,m})/2)$$

para cada $i = 1, 2, \dots, m$.

$$(2.10) \quad k_{4,i} = h \cdot f_i(t_j + h, w_{1,j} + k_{3,1}, w_{2,j} + k_{3,2}, \dots, w_{m,j} + k_{3,m}),$$

para cada $i = 1, 2, \dots, m$

y entonces

$$(2.11) \quad w_{i,j+1} = w_{i,j} + 1/6(k_{1,i} + 2 \cdot k_{2,i} + 2 \cdot k_{3,i} + k_{4,i}),$$

para cada $i = 1, 2, \dots, m$.

Observemos que $k_{1,1}, k_{1,2}, \dots, k_{1,m}$ debe calcularse antes de que $k_{2,1}$ sea determinada. El algoritmo siguiente ejecuta el método de Runge – Kutta de cuarto orden para sistema de problemas de valores iniciales.

Algoritmo de Runge- Kutta para sistema de Ecuaciones Diferenciales Ordinarias de primer orden.

Para aproximar la solución de sistema de ecuaciones diferenciales de primer orden de problemas de valor inicial de primer orden.

$$u'_j = f_j(t, u_1, u_2, \dots, u_m), \quad j = 1, 2, \dots, m$$

$$a \leq t \leq b, \quad u_j(a) = \alpha_j \quad j = 1, 2, \dots, m$$

en $(N + 1)$ números igualmente espaciados en el intervalo $[a, b]$

ENTRADA: Puntos extremos a, b ; números de ecuaciones m ; entero N ; condiciones iniciales, $\alpha_1, \alpha_2, \dots, \alpha_m$.

Salida: Aproximaciones w_j a $u_j(t)$ en los $(N + 1)$ valores de t .

Paso 1. Tomar $h = (b - a)/N$

$$t = a$$

Paso 2. Para $j = 1, 2, \dots, m$

$$\text{Tomar } w_j = \alpha_j.$$

Paso 3. SALIDA: $(t, w_1, w_2, \dots, w_m)$

Paso 4. Para $i = 1, 2, \dots, n$ seguir los pasos 5 – 11

Paso 5. Para $j = 1, 2, \dots, m$ tomar

$$k_{1,j} = h * f_j(t, w_1, w_2, \dots, w_m)$$

Paso 6. Para $j = 1, 2, \dots, m$ tomar

$$k_{2,j} = h * f_j(t + h/2, w_1 + (k_{1,1})/2, w_2 + (k_{1,2})/2, \dots, w_m + (k_{1,m})/2).$$

Paso 7. Para $j = 1, 2, \dots, m$ tomar

$$k_{3,j} = h * f_j(t + h/2, w_1 + (k_{2,1})/2, w_2 + (k_{2,2})/2, \dots, w_m + (k_{2,m})/2).$$

Paso 8. Para $j = 1, 2, \dots, m$ tomar

$$k_{4,j} = h * f_j(t + h/2, w_1 + k_{3,1}, w_2 + k_{3,2}, \dots, w_m + k_{3,m}).$$

Monografía para optar al título de Licenciatura en Matemática.

Paso 9. Para $j = 1, 2, \dots, m$ tomar

$$w_{i,j+1} = w_{i,j} + (k_{1,j} + 2 * k_{2,j} + 2 * k_{3,j} + k_{4,j})/6;$$

Paso 10. Tomar $t = a + i * h$

Paso 11. SALIDA: $(t, w_1, w_2, \dots, w_m)$

Paso 12. PARAR.

PROGRAM SISTEMA_DE_ECUACIONES_DIFERENCIALES_RUNGE-KUTTA;

(*PROGRAMA PARA RESOLVER SISTEMAS DE ECUACIONES DIFERENCIALES*)

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

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

(* $a \leq t \leq b$; $x(a) = \text{alfa}_1$; $y(b) = \text{alfa}_2$ *)

USES CRT;

CONST MAX = 1500;

VAR

w1, w2, t, x, y: ARRAY [0.. MAX] OF REAL;

n, j, I, r, m, s: INTEGER;

k11, k12, k21, k22, k31, k32, k41, k42, C1, C2, a, b, h: REAL;

FUNCTION F (m: INTEGER; t, x, y: REAL): REAL;

BEGIN

CASE m OF

1:F: = - 4*x + 3*y + 6;

2:F: = x - 4*y;

3:F: = 2*x + 2*y;

END;

END;

FUNCTION G(r: INTEGER; t, x, y: REAL): REAL;

BEGIN

CASE r OF

1: G: = - 2.4 * x + 1.6 *y + 3.6;

2: G: = 3 * x + y;

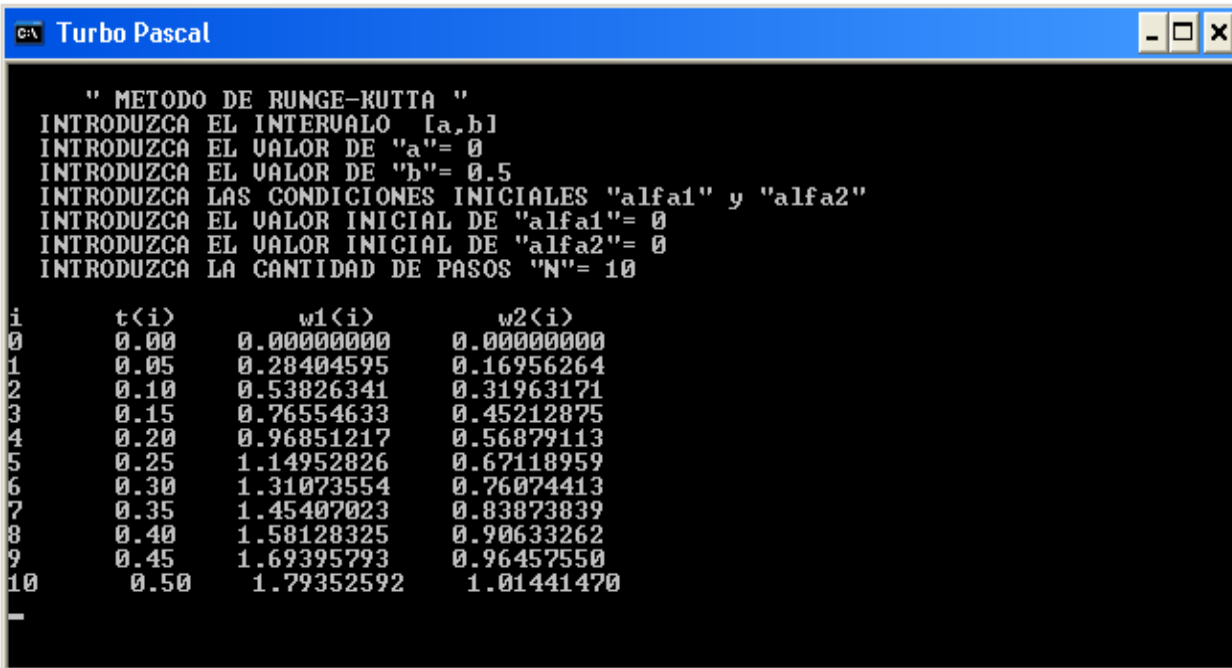
```
3: G: = x + y;
END;
END;
BEGÍN
CLRSCR;
WRITELN ('ELIJA UNA FUNCION...');
WRITELN ('1:F(x, y, t) = - 4* x + 3 * y + 6 ');
WRITELN ('2:F(x, y, t) = x - 4 * y ');
WRITELN ('3: F(x, y, t) = 2* x + 2 * y ');
WRITELN;
WRITELN ('ESCOJA EL NUMERO DE LA FUNCION...');
READLN(m);
CLRSCR;
WRITELN ('ELIJA UNA FUNCION...');
WRITELN ('1: G(x, y, t) = - 2.4 * x + 1.6 * y + 3.6 ');
WRITELN ('2: G(x, y, t) = 3 * x + y ');
WRITELN ('3: G(x, y, t) = x + y ');
WRITELN;
WRITELN ('ESCOJA EL NUMERO DE LA FUNCION...');
READLN (r);
CLRSCR;
WRITELN ('INTRODUZCA EL INTERVALO [a, b]');
WRITE ('INTRODUZCA EL VALOR DE "a" = ');
READLN (a);
```


Monografía para optar al título de Licenciatura en Matemática.

```
WRITE ('INTRODUZCA EL VALOR DE "b" = ');
READLN (b);
WRITELN('INTRODUZCA LAS CONDICIONES INICIALES "alfa1" y "alfa2"
');
WRITE ('INTRODUZCA EL VALOR INICIAL DE "alfa1" = ');
READLN (C1);
WRITE ('INTRODUZCA EL VALOR INICIAL DE "alfa2" = ');
READLN (C2);
WRITE ('INTRODUZCA LA CANTIDAD DE PASOS "N" = ');
READLN (N);
h: = (b - a)/N;
t[0]: = a;
w 1[0]: = C1;
w 2[0]: = C2
WRITELN ('i t (i), w1(i) w2(i) ');
WRITELN (0:0, t[i]:10:2, ' ,w1[1]:4:8', ' ,w2[i]:4:8);
FOR i: = 1 TO N DO
BEGIN
k11: = h*F(m, t[i-1], w1[i-1], w2[i-1]);
k12: = h*G(r, t[i-1], w1[i-1], w2[i-1]);
k21: = h *F(m, t[i-1] + h/2, w1[i-1] + (k11)/2, w2[i-1] + (k12)/2);
k22: = h*G(r, t[i-1] + h/2, w1[i-1] + (k11)/2, w2[i-1] + (k12)/2);
k31: = h*F(m, t[i-1] + h/2, w1[i-1] + (k21)/2, w2[i-1] + (k22)/2);
```

```
k32:= h*G(r, t[i-1] + h/2, w1[i-1] + (k21)/2, w2[i-1] + [k22)/2);
k41:= h*F(m, t[i-1] + h,w1[i-1] + k31, w2[i-1] + k32);
k42:= h*G(r, t[i-1] + h, w1[i-1] + k31, w2[i-1] + k32);
w1[i]:= w1[i-1] + (k11 + 2 *(k21) + 2 *(k31) + k41)/6;
w2[i]:= w2[i-1] + (k12 + 2* (k22) + 2* (k32) + k42)/6;
t[i]:= a + i*h;
WRITELN (i, t[i]:10:2,' ',w1[i]:4:6,' ',w2[i]:4:6);
END;
REPEAT UNTIL KEYPRESSED;
END.
```

A continuación presentamos una captura de la salida del programa de Runge-Kutta en Turbo Pascal (versión 7).



```
" METODO DE RUNGE-KUTTA "
INTRODUZCA EL INTERVALO [a,b]
INTRODUZCA EL VALOR DE "a"= 0
INTRODUZCA EL VALOR DE "b"= 0.5
INTRODUZCA LAS CONDICIONES INICIALES "alfa1" y "alfa2"
INTRODUZCA EL VALOR INICIAL DE "alfa1"= 0
INTRODUZCA EL VALOR INICIAL DE "alfa2"= 0
INTRODUZCA LA CANTIDAD DE PASOS "N"= 10

i      t(i)      w1(i)      w2(i)
0      0.00      0.00000000  0.00000000
1      0.05      0.28404595  0.16956264
2      0.10      0.53826341  0.31963171
3      0.15      0.76554633  0.45212875
4      0.20      0.96851217  0.56879113
5      0.25      1.14952826  0.67118959
6      0.30      1.31073554  0.76074413
7      0.35      1.45407023  0.83873839
8      0.40      1.58128325  0.90633262
9      0.45      1.69395793  0.96457550
10     0.50      1.79352592  1.01441470
-
```

TABLA #2 RUNGE-KUTTA

T(i)	W1(i)	W2(i)	x1(i)	y1(i)	$\ w1 - x1\ $	$ w2 - y2 $
0.00	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
0.05	0.28404595	0.16956264	0.28404623	.16956282	$1.63 \cdot 10^{-6}$	$1.8 \cdot 10^{-7}$
0.10	0.53826341	0.31963171	0.53826391	0.31963204	$5 \cdot 10^{-7}$	$3.3 \cdot 10^{-7}$
0.15	0.77554633	0.45212875	0.76554701	0.45212920	$6.8 \cdot 10^{-7}$	$4.5 \cdot 10^{-7}$
0.20	0.96851217	0.56879113	0.96851299	0.56879168	$8.2 \cdot 10^{-7}$	$5.5 \cdot 10^{-7}$
0.25	1.14952826	0.67118959	1.14952918	0.67119021	$9.2 \cdot 10^{-7}$	$6.2 \cdot 10^{-7}$
0.30	1.31073554	0.76074413	1.31073655	0.76074480	$1.10 \cdot 10^{-6}$	$6.7 \cdot 10^{-7}$
0.35	1.45407023	0.83873839	1.45407129	0.83873910	$1.01 \cdot 10^{-6}$	$7.1 \cdot 10^{-7}$
0.40	1.58128325	0.90633262	1.58128435	0.90633336	$1.1 \cdot 10^{-6}$	$7.4 \cdot 10^{-7}$
0.45	1.69395793	0.96457550	1.69395904	0.96457624	$1.11 \cdot 10^{-6}$	$7.5 \cdot 10^{-7}$
0.50	1.79352592	1.01441470	1.79352705	1.01441545	$1.13 \cdot 10^{-6}$	$1.13 \cdot 10^{-6}$

METODO PREDICTOR-CORRECTOR (ADAMS-BASHFORTH Y ADAMS-MOULTON)

Los métodos de Euler y Runge-Kutta se llaman métodos de un paso debido a que la aproximación para el punto de malla t_{i+1} involucran solo información de uno de los puntos de mallas anteriores t_i . Aunque estos métodos generalmente utilizan información de la evaluación de la función entre los puntos t_i y t_{i+1} no retienen esta información para usarla directamente en aproximaciones futuras. Toda la información utilizada por estos métodos se obtiene dentro del intervalo sobre el cual la solución se está aproximando.

Puesto que se cuenta con la solución aproximada en cada punto de malla t_0, t_1, \dots, t_i . Antes de obtener la aproximación en t_{i+1} y debido a que el error $|w_j - y(t_j)|$ tiende a un incremento con j , parece razonable desarrollar métodos que usen estos datos anteriores más precisos cuando se aproxime la solución en t_{i+1} .

Los métodos que utilizan la aproximación en más de unos de los puntos anteriores de malla para determinar la aproximación en los puntos siguientes se llaman **método multipaso**. La definición de estos dos tipos de método de multipaso se proporciona a continuación.

Definición:

Un método multipaso para resolver el problema de valor inicial.

$$y' = f(t, y), \quad a \leq t \leq b \quad y(a) = \alpha \quad (2.12)$$

Es una cuya ecuación en diferencia para encontrar la aproximación w_{i+1} en el punto de malla t_{i+1} , puede representarse con la siguiente ecuación, donde m es un entero mayor que 1:

$$w_{i+1} = a_{m-1}w_i + a_{m-2}w_{i-1} + \dots + a_0w_{i+1-m} + h[b_m f(t_{i+1}, w_{i+1}) + b_{m-1}f(t_i, w_i) + \dots + b_0 f(t_{i+1-m}, w_{i+1-m})] \quad (2.13)$$

Para $i = m - 1, m, \dots, n-1$, donde los valores iniciales

$$w_0 = \alpha_0, w_1 = \alpha_1, \dots, w_{m-1} = \alpha_{m-1} \text{ están especificados y } h = \frac{b-a}{N}.$$

Cuando $b_m = 0$, el método se llama *explícito* o abierto puesto que la ecuación (2.13) proporciona w_{i+1} explícitamente en términos de valores previamente determinados. Cuando $b_m \neq 0$ el método se llama *implícito* o cerrado puesto que w_{i+1} se presenta cuando ambos lados de la ecuación (2.13) se determina de una manera implícita.

A continuación se dan algunos de los métodos multipasos explícitos juntos con los valores iniciales requeridos y los errores de truncamiento local.

Método de Adams-Bashforth de dos pasos:

$$w_0 = \alpha_0 \quad w_1 = \alpha_1$$

$w_{i+1} = w_i + \frac{h}{2}[3f(t_i, w_i) - f(t_{i-1}, w_{i-1})]$, donde $i = 1, 2, \dots, N-1$, el error de

truncamiento local $\tau_{i+1}(h) = \frac{5}{12} y''''(\mu_i)h^2$, para alguna $\mu_i \in (t_{i-1}, t_{i+1})$.

Método de Adams-Bashforth de tres pasos:

$$w_0 = \alpha_0, \quad w_1 = \alpha_1, \quad w_2 = \alpha_2$$

$w_{i+1} = w_i + \frac{h}{12}[23f(t_i, w_i) - 16f(t_{i-1}, w_{i-1}) + 5f(t_{i-2}, w_{i-2})]$, donde $i = 2, 3, \dots, N-1$

Monografía para optar al título de Licenciatura en Matemática.

El error de truncamiento local es $\tau_{i+1}(h) = \frac{3}{8} y^{(4)}(\mu_i)h^3$, para alguna

$$\mu_i \in (t_{i-2}, t_{i+1})$$

Método de Adams-Bashforth de cuatro pasos:

$$w_0 = \alpha_1, \quad w_1 = \alpha_1, \quad w_2 = \alpha_2, \quad w_3 = \alpha_3.$$

$$w_{i+1} = w_i + \frac{h}{24} [55f(t_i, w_i) - 59f(t_{i-1}, w_{i-1}) + 37f(t_{i-2}, w_{i-2}) - 9f(t_{i-3}, w_{i-3})] \quad (2.14)$$

Donde $i = 3, 4, \dots, N-1$. El error de truncamiento local es:

$$\tau_{i+1}(h) = \frac{251}{720} y^{(5)}(\mu_i)h^4, \text{ para alguna } \mu_i \in (t_{i-3}, t_{i+1}).$$

Presentaremos algunos de los métodos implícitos más comunes:

Método de Adams-Moulton de dos pasos:

$$w_0 = \alpha_0, \quad w_1 = \alpha_1$$

$$w_{i+1} = w_i + \frac{h}{12} [5f(t_{i+1}, w_{i+1}) + 8f(t_i, w_i) - f(t_{i-1}, w_{i-1})], \text{ donde } i = 1, 2, \dots, N-1, \text{ El}$$

error de truncamiento local $\tau_{i+1}(h) = -\frac{h}{12} y^{(5)}(\mu_i)h^4$, para alguna $\mu_i \in (t_{i-1}, t_{i+1})$.

Método de Adams-Moulton de tres pasos:

$$w_0 = \alpha, \quad w_1 = \alpha_1, \quad w_2 = \alpha_2$$

$$w_{i+1} = w_i + \frac{h}{24} [9f(t_{i+1}, w_{i+1}) + 19f(t_{i-1}, w_{i-1}) - 5f(t_{i-1}, w_{i-1}) + f(t_{i-2}, w_{i-2})] \quad (3.15)$$

donde $i = 2, 3, \dots, N-1$

El error de truncamiento local es $\tau_{i+1}(h) = -\frac{19}{720} y^{(5)}(\mu_i)h^4$, para alguna

$$\mu_i \in (t_{i-2}, t_{i+1})$$

En la práctica, no se emplean los métodos multipasos implícitos de forma aislada. Solo se usan para mejorar aproximaciones obtenidas mediante los métodos explícitos. Lo

recomendable es la combinación de una técnica explícita con una implícita, a la que se le llama Método predictor-corrector. El método explícito predice y el método implícito corrige.

La técnica es la siguiente: El primer paso es calcular los valores iniciales w_0 , w_1 , w_2 y w_3 para el método de Adams-Bashforth de cuatro pasos. Para eso usamos un método de cuarto orden de un paso, el método de Runge-Kutta de cuarto orden. El siguiente paso es calcular una aproximación, $w_4^{(0)}$ para $y(t_4)$ usando el método el método de Adams-Bashforth como predictor:

$$w_4^{(0)} = w_3 + \frac{h}{24} [55f(t_3, w_3) - 59f(t_2, w_2) + 37f(t_1, w_1) - 9f(t_0, w_0)]$$

Esta aproximación se mejora insertando $w_4^{(0)}$ en el segundo miembro del método de Adams-Moulton de tres pasos utilizando este método como corrector:

$$w_4^{(1)} = w_3 + \frac{h}{24} [9f(t_4, w_4^{(0)}) + 19f(t_3, w_3) - 5f(t_2, w_2) + f(t_1, w_1)]$$

El valor de $w_4^{(1)}$ se usa después como una aproximación para $y(t_4)$ y se repite la técnica utilizada en el método de Adams-Bashforth como predictor y el método de Adams-Moulton como corrector para encontrar $w_5^{(0)}$ y $w_5^{(1)}$ que son las aproximaciones inicial y final a $y(t_5)$, etc. En teoría, se pueden obtener aproximaciones mejoradas para $y(t_{i+1})$ iterando la formula de Adams-Moulton

$$w_{i+1}^{(k+1)} = w_i + \frac{h}{24} [9f(t_{i+1}, w_{i+1}^{(k)}) + 19f(t_i, w_i) - 5f(t_{i-1}, w_{i-1}) + f(t_{i-2}, w_{i-2})]$$

En la practica puesto $\{ w_{i+1}^{(k+1)} \}$ converge en la aproximación real dada por la formula implícita y no en la solución $y(t_{i+1})$, es más eficiente usar una reducción en el tamaño de paso cuando se necesite mejor la precisión.

Así generalizaremos esta técnica hacia los sistemas de ecuaciones diferenciales a través del siguiente algoritmo.

Algoritmo Predictor – Corrector de Adams-Bashforth para Sistema de Ecuaciones Diferenciales Ordinarias de primer orden.

Este algoritmo de Adams-Bashforth de cuarto orden como predictor y en una iteración de Adams-Moulton como corrector, con los valores iniciales obtenido del método de Runge-Kutta de cuarto orden.

Para aproximar la solución al sistema de orden n de problemas de valor inicial de primer orden.

$$u'_j = (t, u_1, u_2, \dots, u_m) \quad j = 1, 2, \dots, m$$

$$a \leq t \leq b, \quad u_j(a) = \alpha_j \quad j = 1, 2, \dots, m$$

en $(N + 1)$ números igualmente espaciados en el intervalo $[a, b]$.

ENTRADA: Puntos extremos a, b ; números de ecuaciones m , entero n ; condiciones iniciales $\alpha_1, \dots, \alpha_m$.

SALIDA: aproximaciones w_j a $u_j(t)$ en los $N + 1$ valores de t .

Paso 1. Tomar $h = (b - a)/N$

$$t_0 = a$$

Paso 2. Para $j = 1, 2, \dots, m$

Tomar

$$w_j = \alpha_j$$

Paso 3. SALIDA:

$$(t, w_1, w_1, \dots, w_m)$$

Paso 4. Para $i = 1, 2, 3$. seguir los pasos 5 - 11

(Calcula los valores iniciales usando el método de Runge-Kutta)

Paso 5. Para $j = 1, 2, \dots, m$

tomar

$$k_{1,j} = h * f_j(t_{i-1}, w_{i-1,1}, w_{i-1,2}, \dots, w_{i-1,m})$$

Paso 6. Para $j = 1, 2, \dots, m$

Tomar

$$k_{2,j} = h * f_j(t_{i-1} + h/2, w_{i-1,1} + (k_{1,1})/2, w_{i-1,2} + (k_{1,2})/2, \dots, w_{i-1,m} + (k_{1,m})/2)$$

Paso 7. Para $j = 1, 2, \dots, m$

Tomar

$$k_{3,j} = h * f_j(t_{i-1} + h/2, w_{i-1,1} + (k_{2,1})/2, w_{i-1,2} + (k_{2,2})/2, \dots, w_{i-1,m} + (k_{2,m})/2)$$

Paso 8. Para $j = 1, 2, \dots, m$

Tomar

$$k_{4,j} = h * f_j(t_{i-1} + h, w_{i-1,1} + k_{3,1}, w_{i-1,2} + k_{3,2}, \dots, w_{i-1,m} + (k_{3,m})/2)$$

Paso 9. Para $j = 1, 2, \dots, m$

Tomar

$$w_{i,j+1} = w_{i,j} + (k_{1,j} + 2 * k_{2,j} + 2 * k_{3,j} + k_{4,j})/6$$

Paso 10. Tomar

$$t_i = a + i * h$$

Paso 11. SALIDA:

$$(t_i, w_{i,1}, w_{i,2}, \dots, w_{i,m})$$

Paso 12. Para $i = 4, \dots, N$ seguir los pasos 13-18.

Paso 13. Para $j = 1, 2, \dots, m$ tomar

$$t = a + i * h$$

Monografía para optar al título de Licenciatura en Matemática.

(*PREDICE w_i usando Adams-Bashforth*)

$$w_{i,j} = w_{3,j} + h * [55 * f_j(t_3, w_{3,1}, w_{3,2}, \dots, w_{3,m}) - 59 * f_j(t_2, w_{2,1}, w_{2,2}, \dots, w_{2,m}) + 37 * f_j(t_1, w_{1,1}, w_{1,2}, \dots, w_{1,m}) - 9 * f_j(t_0, w_{0,1}, w_{0,2}, \dots, w_{0,m})] / 24;$$

Paso 14. Para $j = 1, 2, \dots, m$

Tomar

(*CORRIGE w_i usando Adams-Moulton*)

$$w_{i,j} = w_{3,j} + h * [9 * f_j(t_i, w_{i,1}, w_{i,2}, \dots, w_{i,m}) + 19 * f_j(t_3, w_{3,1}, w_{3,2}, \dots, w_{3,m}) - 5 * f_j(t_2, w_{2,1}, w_{2,2}, \dots, w_{2,m}) + f_j(t_1, w_{1,1}, w_{1,2}, \dots, w_{1,m})] / 24;$$

Paso 15 SALIDA

$$(t, w_{i,1}, w_{i,2}, \dots, w_{i,m})$$

Paso 16. Para $j = 0, 1, 2$.

Paso 17. Para $j = 1, 2, \dots, m$

Tomar $t_j = t_{j+1}$; (*Se prepara para la siguiente iteración*)

$$w_j = w_{j+1}$$

Paso 18. Tomar $t_3 = t$

$$w_3 = w_j$$

Paso 19. PARAR

A continuación presentamos el programa elaborado para el algoritmo de este método.

**PROGRAM PREDICTOR_CORRECTOR_DE_ADAMS_BASHFORTH;
(*PROGRAMA PARA RESOLVER SISTEMAS DE ECUACIONES**

DIFERENCIALES*)

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

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

USES CRT;

CONST MAX = 1500;

VAR

w1, w2, t, x, y: ARRAY [0..MAX] OF REAL;

n, i, j, r, m: INTEGER;

K11, k12, k21, k22, k31, k32, k41, k42, C1, C2, a, b, h: REAL;

FUNCTION F(m: INTEGER, t, x, y: REAL): REAL;

BEGIN

CASE m OF

1:F: = $-4*x + 3*y + 6$.

2:F: = $x - 4*y$;

3:F: = $2*x + 2*y$;

END;

END;

FUNCTION G(r: INTEGER; t, x, y: REAL):REAL;

BEGIN

CASE r OF

1:G: = $-2.4*x + 1.6*y + 3.6$;

2:G: = $x + y$;

3:G: = $3*x + y$;

END;

END;

```
BEGIN
CLRSCR;
WRITELN ('ELIJA UNA FUNCION...');
WRITELN ('1: F(x, y, t) = - 4*x + 3*y + 6 ');
WRITELN ('2: F(x, y, t) = x - 4*y ');
WRITELN('3: F(x, y, t) = 2*x + 2* y ');
WRITELN;
WRITELN ('ESCOJA EL NUMERO DE LA FUNCION...');
READLN (m);
CLRSCR;
WRITELN ('ELIJA UNA FUNCION');
WRITELN ('1:G(x,y,t) = -2.4*x + 1.6*y + 3.6 ');
WRITELN ('2:G(x, y, t) = x + y ');
WRITELN ('3:G(x,y,t) = 3*x + y ');
WRITELN;
WRITELN ('ESCOJA EL NUMERO DE LA FUNCION....');
READLN (n);
CLRSCR;
WRITELN ('INTRODUZCA EL INTERVALO [a, b]');
WRITE ('INTRODUZCA EL VALOR DE "a" = ');
READLN (a);
WRITE ('INTRODUZCA EL VALOR DE "b" = ');
READLN (b);
WRITELN ('INTRODUZCA LAS CONDICIONES INICIALES "alfa1" y "alfa2"
');
WRITE ('INTRODUZCA EL VALOR INICIAL DE "alfa1" = ');
READLN (C1);
```

Monografía para optar al título de Licenciatura en Matemática.

```
WRITE ('INTRODUZCA EL VALOR INICIAL DE "alfa2" = ');
READLN (C2);
WRITE ('INTRODUZCA LA CANTIDAD DE PASOS "N" = ');
h: = (b - a)/N;
t[0]: = a;
w1[0]: = C1;
w2[0]: = C2;
WRITELN ('i   t(i)       w1(i)       w2(i) ');
WRITELN (0:0, t[i]:10:2,' ',w1[i]:4:8,' ',w2[i]:4:8);
(* Runge-Kutta *)
FOR i: = 1 TO 3 DO
BEGIN
k11: = h*F(m, t[i-1], w1[i-1], w2[i-1]);
k12: = h*G(r, t[i-1], w1[i-1], w2[i-1]);
k21: = h*F(m, t[i-1] + h/2, w1[i-1] + (k11)/2, w2[i-1] + (k12)/2);
k22: = h*G(r, t[i-1] + h/2, w1[i-1] + (k11)/2, w2[i-1] + (k12)/2);
k31: = h*F(m, t[i-1] + h/2, w1[i-1] + (k21)/2, w2[i-1] + (k22)/2);
k32: = h*G(r, t[i-1] + h/2, w1[i-1] + (k21)/2, w2[i-1] + (k22)/2);
k41: = h*F(m, t[i-1] + h, w1[i-1] + k31, w2[i-1] + k32);
k42: = h*G(r, t[i-1] + h, w1[i-1] + k31, w2[i-1] + k32);
w1 [i] := w1[i-1] + (k11 + 2*(k21) + 2*(k31) + k41)/6;
w2 [i]: = w2[i-1] + (k12 + 2*(k22) + 2*(k32) + k42)/6;
t[i]: = a + i*h;
WRITELN (i, t[i]:10:2,' ',w1[i]:4:8,' ',w2[i]:4:8);
END;
FOR i: = 4 TO N DO
BEGIN
```

```
BEGIN
t[i]: = a + i*h;
BEGIN
(* PREDICE Adams-Bashforh *)
w1[i]: = w1[3] + h* (55*F(m, t[3], w1[3],w2[3]) – 59*F(m, t[2],
w1[2], w2[2]) + 37*F (m, t[1], w1[1], w2[1])- 9* F(m, t[0], w1[0], w2[0]))/24;
w2[i]: = w2[3]+ h* (55* G(r, t[3], w1[3],w2[3]) – 59* G(r, t[2],
w1[2], w2[2]) + 37*G(r, t[1], w1[1], w2[1]) – 9* G(r, t[0], w1[0],w2[0]))/24;
(* CORRIGE Adams-Moulton *)
w1 [i]: = w 1[3] + h* (9* F(m, t[i], w1[i], w2[i]) + 19* F(m, t[3],w1[3], w2[3]) –
5*F(m,t [2], w1[2], w2[2]) + F(m, t[1], w1[1], w2[1]))/24;
w2[i]: = w2[3] + h* (9* G(r, t[i], w1[i], w2[i]) + 19* G(r, t[3], w[3], w2[3] –
5*G(r, t[2], w1[2], w2[2]) + G(r, t[1], w1[1], w2[1]))/24;
END;
(*SALIDA*)
WRITELN (i, t[i]:10:2,’ ‘,w1[i]:4:8,’ ‘,w2[i]:4:8);
END;
FOR j: = 0 TO 2 DO
BEGIN
t[j]: = t [j + 1];
w 1[j]: = w1[j + 1];
W2 [j]: = w2[j +1];
END;
BEGIN
t[3]: = t[i];
w1[3]: = w1[i];
w2[3]: = w2[i];
```

Monografía para optar al título de Licenciatura en Matemática.

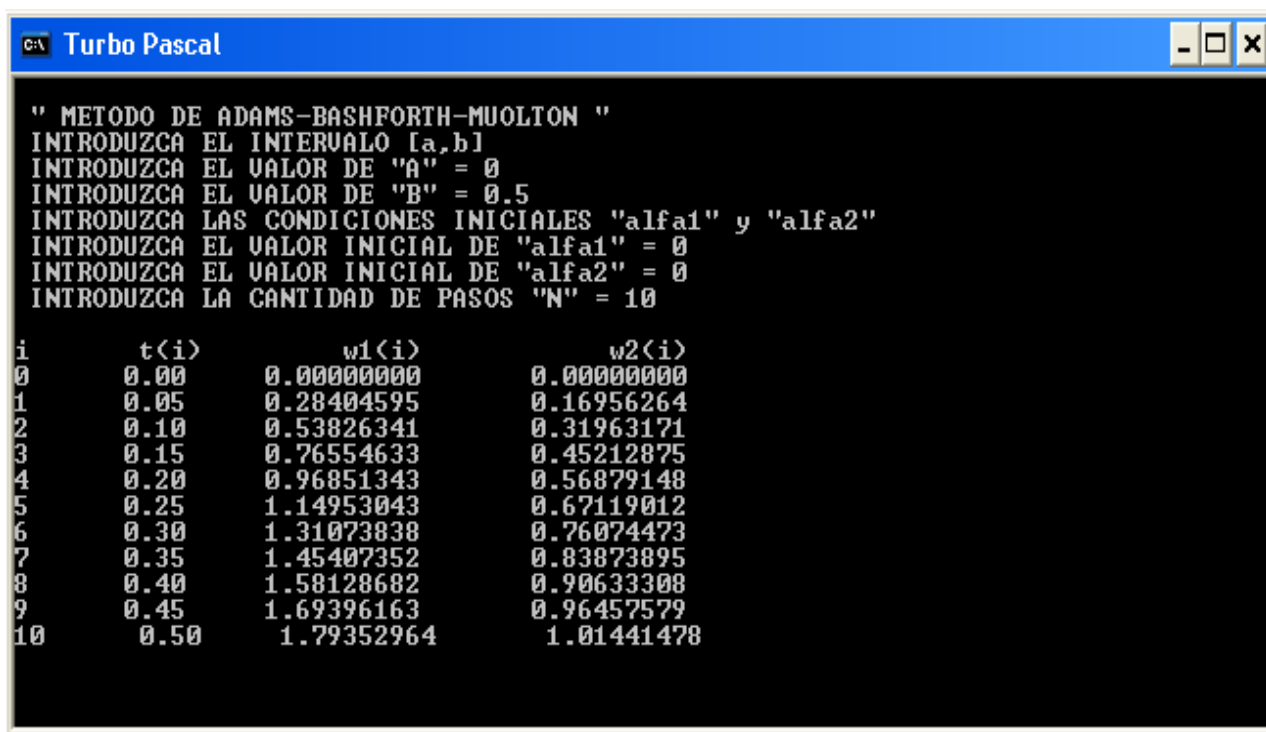
END;

END;

REPEAT UNTIL KEYPRESSED;

END.

A continuación presentamos una captura de la salida del programa de *Adams-Bashforth-Muolton* en Turbo Pascal (versión 7).



```
" METODO DE ADAMS-BASHFORTH-MUOLTON "  
INTRODUZCA EL INTERVALO [a,b]  
INTRODUZCA EL VALOR DE "A" = 0  
INTRODUZCA EL VALOR DE "B" = 0.5  
INTRODUZCA LAS CONDICIONES INICIALES "alfa1" y "alfa2"  
INTRODUZCA EL VALOR INICIAL DE "alfa1" = 0  
INTRODUZCA EL VALOR INICIAL DE "alfa2" = 0  
INTRODUZCA LA CANTIDAD DE PASOS "N" = 10  
  
i      t<i>      w1<i>      w2<i>  
0      0.00      0.00000000  0.00000000  
1      0.05      0.28404595  0.16956264  
2      0.10      0.53826341  0.31963171  
3      0.15      0.76554633  0.45212875  
4      0.20      0.96851343  0.56879148  
5      0.25      1.14953043  0.67119012  
6      0.30      1.31073838  0.76074473  
7      0.35      1.45407352  0.83873895  
8      0.40      1.58128682  0.90633308  
9      0.45      1.69396163  0.96457579  
10     0.50      1.79352964  1.01441478
```

TABLA # 3 Adams-Bashforth-Moulton.

t(i)	W1(i)	w2(i)	X1(i)	y1(i)	w1 - x1	w2 - y2
0.00	0.00000000	0.00000000	0.00000000	0.00000000	0.000000	0.000000
0.05	0.28404595	0.16956264	0.28404623	0.16956282	$2.8 \cdot 10^{-7}$	$1.8 \cdot 10^{-7}$
0.10	0.53826341	0.31963171	0.53826391	0.31963204	$5 \cdot 10^{-7}$	$3.3 \cdot 10^{-7}$
0.15	0.76554633	0.45212875	0.76554701	0.45212920	$6.8 \cdot 10^{-7}$	$4.5 \cdot 10^{-7}$
0.20	0.96851343	0.56879148	0.96851299	0.56879168	$4.4 \cdot 10^{-7}$	$2 \cdot 10^{-7}$
0.25	1.14953043	0.67119012	1.14952918	0.67119021	$1.25 \cdot 10^{-6}$	$9 \cdot 10^{-8}$
0.30	1.31073838	0.76074473	1.31073655	0.76074480	$1.83 \cdot 10^{-6}$	$7 \cdot 10^{-8}$
0.35	1.45407352	0.83873895	1.45407129	0.83873910	$2.23 \cdot 10^{-6}$	$1.5 \cdot 10^{-7}$
0.40	1.58128682	0.90633308	1.58128435	0.9063336	$2.47 \cdot 10^{-6}$	$2.8 \cdot 10^{-7}$
0.45	1.69396163	0.96457579	1.69395904	0.96457624	$2.59 \cdot 10^{-6}$	$4.5 \cdot 10^{-7}$
0.50	1.79352964	1.01441479	1.79352705	1.01441545	$2.59 \cdot 10^{-6}$	$6.6 \cdot 10^{-7}$

METODO PREDICTOR-CORRECTOR DE MILNE-SIMPSON PREDICTOR-CORRECTOR DE MILNE-SIMPSON

Este método de multipaso es también combinación de una técnica explícita con una implícita llamado corrector-predictor.

-Donde el método *predictor* es llamado **Método de Milne:**

$$w_0 = \alpha \quad w_1 = \alpha_1, w_2 = \alpha_2, w_3 = \alpha_3.$$

$$w_{i+1} = w_{i-3} + \frac{4h}{3} [2f(t_i, w_i) - f(t_{i-1}, w_{i-1}) + 2f(t_{i-2}, w_{i-2})] \quad (3.16)$$

Para $i = 3, 4, \dots, N - 1$. El cual tiene un error de truncamiento local: $\frac{14}{15}h^4 \cdot y^{(5)}(\xi)$ para alguna $\xi \in (t_{i-3}, t_{i+1})$.

-El método corrector es llamado **Metodo de Simpson:**

$$w_0 = \alpha, w_1 = \alpha_1, w_2 = \alpha_2$$

$$w_{i+1} = w_{i-1} + \frac{h}{3} [f(t_{i+1}, w_{i+1}) + 4f(t_i, w_i) + f(t_{i-1}, w_{i-1})] \quad (3.17)$$

Para $i = 3, 4, \dots, N - 1$. El cual tiene un error de truncamiento local: $-\frac{h^4}{90} \cdot y^{(5)}(\xi)$ para alguna $\xi \in (t_{i-1}, t_{i+1})$.

La técnica es la siguiente: El primer paso es calcular los valores iniciales w_0, w_1, w_2 y w_3 para el método de Milne de cuatro pasos. Para eso usamos un método de cuarto orden de un paso, el método de Runge-Kutta de cuarto orden. El siguiente paso es calcular una aproximación, $w_4^{(0)}$ para $y(t_4)$ usando el método el método de Milne como predictor:

Monografía para optar al título de Licenciatura en Matemática.

$$w_4^{(0)} = w_0 + \frac{4h}{3} [2f(t_3, w_3) - f(t_2, w_2) + 2f(t_1, w_1)]$$

Esta aproximación se mejora insertando $w_4^{(0)}$ en el segundo miembro del método de Simpson de tres pasos utilizando este método como corrector:

$$w_4^{(1)} = w_2 + \frac{h}{3} [f(t_4, w_4^{(0)}) + 4f(t_3, w_3) + f(t_2, w_2)]$$

El valor de $w_4^{(1)}$ se usa después como una aproximación para $y(t_4)$ y se repite la técnica utilizada en el método de Milne predictor y el método de Simpson como corrector para encontrar $w_5^{(0)}$ y $w_5^{(1)}$ que son las aproximaciones inicial y final a $y(t_5)$, etc.

Así generalizaremos esta técnica hacia los sistemas de ecuaciones diferenciales a través del siguiente algoritmo.

Algoritmo Predictor – Corrector de Milne-Simpson para Sistema de Ecuaciones Diferenciales de primer orden.

Al igual que los métodos estudiados anteriormente también este método es una generación del método corrector de Milne-Simpson para una sola ecuación de primer orden.

El algoritmo para aproximar la solución al sistema de orden m de problemas de valor inicial de primer orden es:

$$u'_j = f_j(t, u_1, u_2, \dots, u_m) \quad j = 1, 2, \dots, m$$

$$a \leq t \leq b \quad u_j(a) = \alpha_j \quad j = 1, 2, \dots, m$$

en $(N + 1)$ números igualmente espaciados en el intervalo $[a, b]$

ENTRADA: Puntos extremos a, b ; números de ecuaciones m ; entero N , condiciones iniciales $\alpha_1, \dots, \alpha_m$

SALIDA: aproximaciones w_j a $u_j(t)$ en los $(N + 1)$ valores de t .

Paso 1. Tomar $h = (b - a)/N$

$$t_0 = a$$

Paso 2. Para $j = 1, 2, \dots, m$

$$\text{Tomar } w_j = \alpha_j$$

Monografía para optar al título de Licenciatura en Matemática.

Paso 3. SALIDA:

$$(t, w_1, w_2, \dots, w_m)$$

Paso 4. Para $i = 1, 2, 3$. Seguir los pasos 5 – 11.

(Calcula los valores iniciales usando el método de Runge-Kutta)

Paso 5. Para $j = 1, 2, \dots, m$

Tomar

$$k_{1,j} = h * f_j(t_{i-1}, w_{i-1,1}, w_{i-1,2}, \dots, w_{i-1,m})$$

Paso 6. Para $j = 1, 2, \dots, m$

Tomar

$$k_{2,j} = h * f_j(t_{i-1} + h/2, w_{i-1,1} + (k_{1,1})/2, w_{i-1,2} + (k_{1,2})/2, \dots, w_{i-1,m} + k_{1,m})/2.$$

Paso 7. Para $j = 1, 2, \dots, m$

Tomar

$$k_{3,j} = h * f_j(t_{i-1} + h/2, w_{i-1,1} + (k_{2,1})/2, w_{i-1,2} + (k_{2,2})/2, \dots, w_{i-1,m} + (k_{2,m})/2)$$

Paso 8. Para $j = 1, 2, \dots, m$

Tomar

$$k_{4,j} = h * f_j(t_{i-1} + h, w_{i-1,1} + k_{3,1}, w_{i-1,2} + k_{3,2}, \dots, w_{i-1,m} + (k_{3,m})/2)$$

Paso 9. Para $j = 1, 2, \dots, m$

Tomar

$$w_{i,j+1} = w_{i,j} + k_{1,j} + 2 * k_{2,j} + 2 * k_{3,j} + k_{4,j})/6$$

Paso 10. Tomar

$$t_i = a + i * h$$

Paso 11. SALIDA

$$(t_i, w_{i,1}, w_{i,2}, \dots, w_{i,m})$$

Monografía para optar al título de Licenciatura en Matemática.

Paso 12. Para $i = 4, \dots, N$ seguir los pasos 13 – 18

Paso 13. Para $j = 1, 2, \dots, m$

Tomar

$$t = a + i \cdot h$$

(*PREDICE w_i usando *Milne* *)

$$w_{i,j} = w_{0,j} + 4h \cdot (2 \cdot f_j(t_3, w_{3,1}, w_{3,2}, \dots, w_{3,m}) - f_j(t_2, w_{2,1}, w_{2,2}, \dots, w_{2,m}) + 2f_j(t_i, w_{i,1}, w_{i,2}, \dots, w_{i,m})) / 3;$$

(*CORRIGE w_i usando *Simpson* *)

$$w_{i,j} = w_{2,1} + h \cdot f_j(t_i, w_{i,1}, w_{i,2}, \dots, w_{i,m}) + 4 \cdot f_j(t_3, w_{3,1}, w_{3,2}, \dots, w_{3,m}) + f_j(t_2, w_{2,1}, w_{2,2}, \dots, w_{2,m}) / 3;$$

Paso 15. SALIDA.

$$(t, w_1, w_2, \dots, w_m)$$

Paso 16. Para $j = 0, 1, 2$.

Paso 17. Para $j = 1, 2, \dots, m$

Tomar

$$t_j = t_{j+1} \quad (\text{Se prepara para la siguiente iteración})$$

$$w_j = w_{j+1}$$

Paso 18. Tomar

$$t_3 = 3$$

$$w_3 = w_1$$

Paso 19. PARAR.

PROGRAM PREDICTOR_CORRECTOR_MILNE_SIMPSON;

**(*PROGRAMA PARA RESOLVER SISTEMAS DE ECUACIONES
DIFERENCIALES*)**

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

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

USES CRT;

CONST MAX = 1500;

VAR

w1, w2, t, x, y: ARRAY [0 .. MAX] OF REAL;

N, i, j, r, m: INTEGER;

k11, k12, k21, k22, k31, k32, k41, k42, C1, C2, a, b, h: REAL;

FUNCTION F(m: INTEGER, t, x, y: REAL): REAL;

BEGIN

CASE m OF

1:F: = - 4* x + 3* y + 6;

2:F: = x - 4*y;

3:F: = 2* x + 2*y;

END;

END;

FUNCTION G(r: INTEGER; t, x, y: REAL): REAL;

BEGIN

CASE r OF

1:G: = - 2.4* x + 1.6* y + 3.6;

2:G: = x + y;

3:G: = 3* x + y;

END;

```
END;
BEGIN
CLRSCR;
WRITELN ('ELIJA UNA FUNCION...');
WRITELN ('1: F(x, y, t) = - 4*x + 3* y + 6 ');
WRITELN ('2: F(x, y, t) = x = 4*y ');
WRITELN ('3: F(x, y, t) = 2* x + 2* y ');
WRITELN;
WRITELN ('ESCOJA EL NUMERO DE LA FUNCION...');
READLN (m);
CLRSCR;
WRITELN ('ELIJA UNA FUNCION');
WRITELN ('1:G(x, y, t) = - 2.4* x + 1.6* y + 3.6 ');
WRITELN ('2:G(x, y, t) = x + y ');
WRITELN ('3:G(x, y, t) = 3* x + y ');
WRITELN;
WRITELN; ('ESCOJA EL NUMERO DE LA FUNCION...');
READLN (N);
CLRSCR;
WRITELN ('INTRODUZCA EL INTERVALO [a, b]');
WRITE ('INTRODUZCA EL VALOR DE "a" = ');
READLN (a);
WRITE ('INTRODUZCA EL VALOR DE "b" = ');
READLN (b);
WRITELN('INTRODUZCA LAS CONDICIONES INICIALES "alfa1 y
"alfa2"');
WRITE ('INTRODUZCA EL VALOR INICIAL DE "alfa1" = ');
```

```
READLN (C1);
WRITE ('INTRODUZCA EL VALOR INICIAL DE "alfa2" = ');
READLN (C2);
WRITE ('INTRODUZCA LA CANTIDAD DE PASOS "n" = ');
h: = (b - a)/N;
t[0]: = a;
w 1[0]: = C1;
w 2[0]: = C2;
WRITELN ('i   t(i)   w1(i)   w2(i) ');
WRITELN (0:0, t[i]:10:2, ' ', w1[i]:4:8, ' ', w2[i]:4:8);
(* Runge-Kutta *)
FOR i: = 1 TO 3 DO
BEGIN
k11: = h* F(m, t[i-1], w1[i-1], w2[i-1]);
k12: = h*G(r,t[i-1], w1[i-1], w2[i-1]
k21: = h*F(m,t[i-1] + h/2, w1[i-1] + (k11)/2, w2[i-1] + (k12)/2);
k22: = h*G(r,t[i-1] + h/2, w1[i-1] + (k11)/2, w2[i-1] + (k12)/2);
k31: = h*F(m,t[i-1] + h/2, w1[i-1] + (k21)/2, w2[i-1] + (k22)/2);
k32: = h*G(r, t[i-1] + h/2, w1[i-1] + (k21)/2, w2[i-1] + (k22)/2);
k41: = h*F(m, t[i-1] + h,w1[i-1] + k31, w2[i-1] + k32);
k42: = h*G(r, t[i-1] + h,w1[i-1] + k31, w2[i-1] + k32);
w1[i]: = w1[i-1] + (k11 + 2*(k21) + 2* (k31) + k41)/6;
w2[i]: = w2[i-1] + (k12 + 2*(K22) + 2* (k32) + k42)/6;
```


Monografía para optar al título de Licenciatura en Matemática.

t[i]: = a + i*h;

WRITELN (i, t[i]: 10:2,' ',w1[i]:4:8,' ',w2[i]:4:8);

END;

FOR i: = 4 TO N DO

BEGIN

 BEGIN

 t[i]: = a + i*h,

 BEGIN

 (*PREDICE Milne *)

 w1[i]: = w1 [0] + 4*h*(2*F(m, t[3], w1[3], w2[3]) – F(m, t[2], w1[2],

 w2[2]) + 2*F(m,t[1], w1[1], w2[1]))/3;

 w2[i]: = w2[0] + 4*h*(2*G(r, t[3], w1[3], w2[3]) –G(r, t[2], w1[2],

 w2[2] + 2* G(r, t[1], w1[1], w2[1]))/3;

 (*CORRIGE Simpson *)

 w1[i]: = w1[2] + h*(F(m,t[i], w1[i], w2[i] + 4*F(m, t[3], w1[3],w2[3]) + F(m, t[2],

 w1[2], w2[2]))/3;

 w2[i]: = w2[2] + h*(G(r,t[i], w1[i], w2[i] + 4*G(r, t[3], w1 [3],w2[3] + G(r, t[2],

 w1[2], w2[2]))/3;

 END;

 (* SALIDA *)

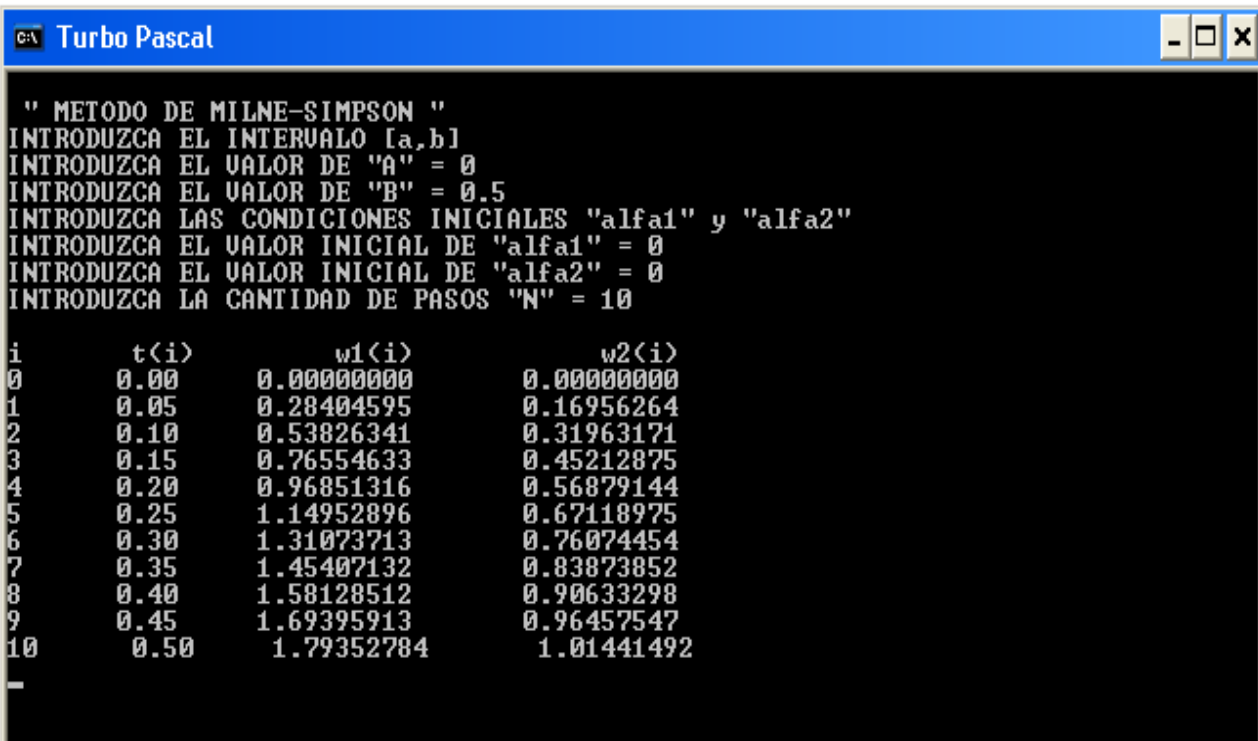
 WRITELN(i, t[i]:10:2,' ',w1[i]:4:8,' ',w2[i]:4:8);

 END;

```
FOR j: = 0 TO 2 DO  
    BEGIN  
        t[j]: = t[j + 1];  
        w1[j]: = w1 [j + 1];  
        w2[j]: = w2[j + 1];  
    END;  
    BEGIN  
t[3]: = t[i];  
        w1[3]: = w1[i];  
        w2[3]: = w2[i];  
    END;  
    END;  
REPEAT UNTIL KEYPRESSED;  
END.
```

Monografía para optar al título de Licenciatura en Matemática.

A continuación presentamos una captura de la salida del programa de *Milne-Simpson* en Turbo Pascal (versión 7).



```
" METODO DE MILNE-SIMPSON "  
INTRODUZCA EL INTERVALO [a,b]  
INTRODUZCA EL VALOR DE "a" = 0  
INTRODUZCA EL VALOR DE "b" = 0.5  
INTRODUZCA LAS CONDICIONES INICIALES "alfa1" y "alfa2"  
INTRODUZCA EL VALOR INICIAL DE "alfa1" = 0  
INTRODUZCA EL VALOR INICIAL DE "alfa2" = 0  
INTRODUZCA LA CANTIDAD DE PASOS "N" = 10  
  
i      t<i>      w1<i>      w2<i>  
0      0.00      0.00000000      0.00000000  
1      0.05      0.28404595      0.16956264  
2      0.10      0.53826341      0.31963171  
3      0.15      0.76554633      0.45212875  
4      0.20      0.96851316      0.56879144  
5      0.25      1.14952896      0.67118975  
6      0.30      1.31073713      0.76074454  
7      0.35      1.45407132      0.83873852  
8      0.40      1.58128512      0.90633298  
9      0.45      1.69395913      0.96457547  
10     0.50      1.79352784      1.01441492  
-
```

TABLA # 4 MÉTODO DE MILNE-SIMPSON

T(i)	w1(i)	w2(i)	X1(i)	y1(i)	 w1 – x1 	 w2 – y2
0.00	0.00000000	0.00000000	0.00000000	0.00000000	0.00000	0.00000
0.05	0.28404595	0.16956264	0.28404623	0.16956282	$2.8 \cdot 10^{-7}$	$1.8 \cdot 10^{-7}$
0.10	0.53826341	0.31963171	0.53826391	0.31963204	$5 \cdot 10^{-7}$	$3.3 \cdot 10^{-7}$
0.15	0.76554633	0.45212875	0.76554701	0.45212920	$6.8 \cdot 10^{-7}$	$4.5 \cdot 10^{-7}$
0.20	0.96851316	0.56879144	0.96851299	0.56879168	$1.7 \cdot 10^{-7}$	$2.4 \cdot 10^{-7}$
0.25	1.14952896	0.67118975	1.14952918	0.67119021	$2.2 \cdot 10^{-7}$	$4.6 \cdot 10^{-7}$
0.30	1.31073713	0.76074454	1.31073655	0.76074480	$5.8 \cdot 10^{-7}$	$2.6 \cdot 10^{-7}$
0.35	1.45407132	0.83873852	1.45407129	0.83873910	$3 \cdot 10^{-7}$	$5.8 \cdot 10^{-7}$
0.40	1.58128512	0.90633298	1.58128435	0.90633336	$7.7 \cdot 10^{-7}$	$3.8 \cdot 10^{-7}$
0.45	1.69395913	0.96457547	1.69395904	0.96457624	$9 \cdot 10^{-8}$	$7.7 \cdot 10^{-8}$
0.50	1.79352784	1.01441492	1.79352705	1.01441545	$7.9 \cdot 10^{-7}$	$5.3 \cdot 10^{-7}$

CONCLUSIONES

1. Los métodos numéricos proporcionan soluciones aproximadas a los Sistemas de Ecuaciones Diferenciales Ordinarias.
2. La vía de las soluciones numéricas en los sistemas de ecuaciones diferenciales debe ser potenciada en aquellos casos en que el uso de las herramientas tradicionales de matemáticas no nos provee de una solución analítica posible o bien esta resulte muy complicada.
3. Podemos concluir de los cuatro programas presentados en este trabajo como son el método de Euler, Runge-Kutta, Predictor-Corrector de Adams-Bashforth y Predictor-Corrector de Milne-Simpson, el más preciso y corrector es de Predictor Corrector de Adams-Bashforth; el cual está combinado con el método de Runge-Kutta para las cuatro primeras iteraciones y a partir de la quinta iteración hasta la n -ésima este predice y corrige cada iteración, mientras que los métodos de Euler y Runge-Kutta son métodos de un pasos, por lo que su convergencia es más lenta que los métodos multipasos. Además los errores absolutos Adams-Bashforth y de Milne-Simpson están comprendidos entre el orden de 10^{-7} y 10^{-8} no así para el método de Euler y Rungr-Kutta sus errores son mayores de 10^{-7} .
4. Por definición Adams-Bashforth es fuertemente estable, no siendo así el método de Milne-Simpson ya que este débilmente estable.
5. Las cifras omitidas introducen un error en los resultados calculados en los programas, este error se llama **error por truncamiento** ya que no podemos usar todos los decimales de cada iteración, ya que sus soluciones están dentro de los números reales. Por lo que quedamos obligados a tomar un número finito de decimales en cada iteración para nuestro caso escogimos ocho cifras decimales.

BIBLIOGRAFÍA

➤ Burden, Richard.

Análisis Numérico, segunda Edición, Grupo Editorial Iberoamérica, S.a. de C.A.
México D.F., 1994.

➤ Boyce Diprima.

Ecuaciones Diferenciales y Problemas con valores en la frontera, Cuarta
Edición, Editorial Limusa, S.A. de C.A. México D.F. 1998.

➤ Denis G. Ziill.

Ecuaciones Diferenciales con Aplicación.
Segunda Edición.

➤ Derrick / Grossman.

Ecuaciones Diferenciales con Aplicaciones.

➤ Nagle / Saff / Zinder.

Ecuaciones Diferenciales y problemas con valores en la frontera
Editorial Adison Wesley Tercera Edición.

➤ Edwards / Penney.

Ecuaciones Diferenciales con Aplicaciones.
PHH Prentice Hall.