

EXPERIENCIA SOBRE RESOLUCIÓN DE UNA ECUACIÓN MEDIANTE PROCESOS ITERATIVOS

**Serrano Gómez, I.
Rojas Matas, A.**

1. INTRODUCCIÓN

Uno de los grandes cambios que se han planteado en los últimos años es la posibilidad de utilizar ordenadores, como un recurso que mejora las posibilidades de resolución de los problemas. Su uso generalizado permite tratar temas que se evitaban por la dificultad de los cálculos a realizar, así como favorecer la posibilidad de insistir en aspectos teóricos.

La finalidad de esta experiencia es usar procedimientos numéricos para resolver ecuaciones. Estos métodos son importantes para los estudiantes de Ingeniería que, en su vida profesional, tratarán con problemas que no siempre se podrán plantear desde un punto de vista analítico: en ocasiones se encontraran que muchos modelos matemáticos no se pueden aplicar directamente a los problemas prácticos, y aunque se pueda encontrar una fórmula explícita que resuelva el problema considerado, otra dificultad puede ser la necesidad de resolver ecuaciones muy complicadas. En consecuencia, será necesario usar modelos más complejos, en los que es conveniente implementar un método numérico.

Entre el software que existe, hemos escogido el Mathematica ya que nos parece un paquete muy interesante que se puede convertir en una herramienta de trabajo adecuada: permite realizar cualquier tipo de operaciones y cálculos (numéricos o simbólicos), se puede trabajar con gráficas y además es interactivo. Por último, Mathematica es también un lenguaje de programación de gran potencia, y de esta forma podemos combinar varias cuestiones metodológicas que centraron nuestra atención al diseñar estas prácticas: capacitar a los estudiantes para resolver problemas relacionados con su especialidad, combinar diferentes estrategias de resolución de un problema para obtener soluciones óptimas (que se aproximen más y con mayor rapidez a la solución del problema); presentar métodos diferentes y poner en evidencia que a priori ninguno es mejor que otro (la eficiencia depende de dos razones fundamen-

mente: la naturaleza del problema a tratar y la decisión sobre lo que en cada caso es una buena aproximación); acercar las matemáticas a la realidad que nos rodea, usando el ordenador como instrumento habitual de trabajo.

Algunas veces es necesario estimar valores iniciales. Esta elección puede influir en la rapidez al encontrar la solución aproximada en menor número de iteraciones. Para decidir qué valores iniciales son adecuados, se pueden combinar métodos iterativos con métodos gráficos, que resultan útiles para obtener una aproximación de la raíz (este puede ser un valor inicial). Mathematica nos permite realizar representaciones gráficas de las funciones consideradas y estimar valores de las raíces con la precisión que se quiera. Esta posibilidad aumenta considerablemente la utilidad de los programas.

El planteamiento seguido en clase ha sido el de combinar explicaciones teóricas, para exponer la base de estos métodos y comprender cómo se deducen las fórmulas utilizadas, con la realización de un programa que nos calcule la raíz de una función en un determinado intervalo.

En la cuestión de resolución numérica de ecuaciones se han utilizado 3 métodos: el de bisección, el de punto fijo y el de Newton-Rhapson. El primero es uno de los métodos que usan intervalos. Aunque en cada uno se sigue una estrategia diferente, en todos se necesita una función que cambie de signo en las cercanías de una raíz de la función. La base está en buscar un método para reducir sistemáticamente el tamaño del intervalo. Los otros son ejemplos de métodos abiertos en los que, partiendo de un valor inicial y usando una fórmula matemática, se calcula una sucesión de números que, a veces, converge a la raíz que se está buscando.

2. MÉTODO DE BISECCIÓN

El método de bisección se toma como primer ejemplo por ser un modelo muy fácil de entender. Consideramos

una ecuación $f(x) = 0$, siendo la función $f(x)$ continua en un intervalo, en el cual la función cambia de signo (así el teorema de Bolzano nos asegura la existencia de una raíz). Si además queremos que esta sea la única raíz, necesitamos que la función sea estrictamente monótona. Para encontrar la raíz, el intervalo se divide siempre en dos. La posición de la raíz se determina situándola en el punto medio del intervalo dentro del cual ocurre el cambio de signo: se evalúa la función en el punto medio, si este valor es cero hemos encontrado la raíz y el proceso se termina; en caso contrario, se selecciona el subintervalo en el cual la función cambie de signo y se repite el procedimiento hasta obtener una aproximación conve-

niente. Este esquema sirve para realizar el programa 1, debatiendo previamente como se puede adoptar un buen criterio de parada: cuando la amplitud del subintervalo con el que se está trabajando sea suficientemente pequeña. Después se ha realizado una mejora del programa (ver programa II), estableciendo otro criterio de parada: lo hará cuando encuentre la solución exacta (en cuyo caso nos dará la solución, nos dirá que es exacta y además nos dará también el número de iteraciones que han sido necesarias), o bien cuando la amplitud del intervalo donde está la solución aproximada sea menor que 10^{-5} (y entonces también nos informará sobre el número de iteraciones que han sido necesarias).

Programa 1: Bisección 1	Programa 2: Bisección 2
<pre> f[x_] := Exp [-x] - x; a = 0.0; b = 2.0; itera = 18; For [i = 1, i <= itera, i++, c = (a + b)/2; If [f[c] == 0, Print ["Solu Exacta ", c]; Break[]]; If [f[c]*f[a] < 0, b = c, a = c];]; If [i == 19, Print ["Solu Aproxim ", c]] </pre>	<pre> f[x_] := Exp [-x] - x; a = 0.0; b = 2.0; itera = 0; amp1 = b - a; tol = 0.00001; while [amp1 > tol, itera = itera + 1; c = (a + b)/2; If [f[c] == 0, Print ["Solu Exacta ", c, "Número itera ", itera]; Break[]]; If [f[c]*f[a] < 0, b = c, a = c]; amp1 = b - a]; If [f[c] != 0, Print ["Solu Aproxim ", c, "Número itera ", itera]]]; </pre>

3. MÉTODO DE PUNTO FIJO

Este método se puede usar para encontrar las raíces de una función $y = f(x)$ que sea continua en un intervalo $[a, b]$. Consiste en modificar la ecuación $f(x) = 0$ (mediante operaciones algebraicas fáciles, o bien añadiendo x a los dos términos), utilizar en su lugar una ecuación equivalente del tipo $g(x) = x$ y buscar un punto fijo. Para asegurarnos que la raíz encontrada es la única que hay en el intervalo, es necesario que g sea una función continua, derivable y además: $|g'(x)| \leq k < 1$ en el intervalo (a, b) . Usando la expresión: $x_{n+1} = g(x_n)$, y partiendo de un valor inicial x_1 , se obtiene una sucesión convergente a la raíz de $f(x) = 0$.

El algoritmo para ordenador es extremadamente fácil. Consiste en un ciclo, que calcula iterativamente nuevas aproximaciones, junto con una declaración lógica, que determina cuando se ha cumplido el criterio de parada.

(Ver Programa III). Sin embargo no siempre se llegará a la raíz, pero cuando se logra, la convergencia es mucho más rápida que en los métodos basados en intervalos.

Programa III: Método de Punto Fijo
<pre> f[x_] := Exp [-x]; act = 0.0; tol = 0.000001; err = 1000.0; itera = 0; maxiter = 20; While [err > tol, itera = itera + 1; If [itera > maxiter, </pre>

```
Print["Efectuado número máximo permitido de
iteraciones, soluc. obtenida ", act]; Break[ ]; ];
nuev = f[act];
Print["Iteración efectuada ", itera , " Solución
obtenida ", nuev];
err = Abs[nuev - act]; act = nuev;]
```

4. MÉTODO DE NEWTON-RAPHSON

El programa IV se ha realizado para el método de Newton-Raphson, que es uno de los más utilizados. Será necesario que la función $f(x)$ sea continua y derivable en el intervalo donde se encuentra la raíz, que además será un número real. Si el valor inicial es x_1 , se puede calcular una tangente a la curva en este punto, y calcular el punto en que dicha recta corta al eje de abscisas. Este valor se toma como una nueva aproximación que mejora a la anterior. Así llegamos a la fórmula de Newton-Raphson:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

que nos dará una sucesión convergente a la raíz de $f(x)$.

Es necesario poner de manifiesto que ahora se pueden encontrar situaciones en las que no es eficiente (por

ejemplo si hay más de una raíz en el intervalo, si hay raíces múltiples, puntos de inflexión de la función en las cercanías de la raíz, o si se encuentran pendientes cercanas o iguales a cero). Para evitar estos problemas, además de combinar con métodos gráficos, es conveniente conocer muy bien la naturaleza del problema que se vaya a considerar.

Programa IV: Método de Newton-Raphson

```
f[x_] := 8 x - Cos[x] - 2 x^2;
act = 200.0;
tol = 0.000001;
err = 10.0;
itera = 0;
While [err > tol, itera = itera + 1;
If [f[act] == 0, Print["Problemas !!!!"]; Break[ ];];
If [f[act] != 0, nuev=act-
f[act]/f'[act]];Print["Iteración efectuada ",itera, "
Soluc. obtenida ", nuev];
err = Abs[nuev-act];
act = nuev;]
```

BIBLIOGRAFÍA

Métodos Numéricos para Ingenieros . Chapra, S.C., Canale R.P.. Editorial McGrawHill.
Matematicas con Mathematica. Ramirez González y otros. Proyecto Sur de Ediciones S.L.