# Tarea No. 3

## La función f(x) = x4 – 3x2 – 3 = 0 tiene una raíz en el intervalo [1, 2]

Utilice los siguientes puntos para aproximar una solución a la raíz en el intervalo dado, con una tolerancia de 10^-5:

Primera transformación**: g(x) = (3x^2 + 3) ^ (1/4)**

P0 = 1.01

Texto

Descripción generada automáticamente

P0 = 1.50

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

P0 = 1.99

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Repita el proceso anterior con una tolerancia de 10^-10:**

P0 = 1.01

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

P0 = 1.50

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

P0 = 1.99

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Segunda transformación: **g(x) = √(3x^2 + 3)∕x**

P0 = 1.01

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

P0 = 1.50

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

P0 = 1.99

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Repita el proceso anterior con una tolerancia de 10^-10:**

P0 = 1.01

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

P0 = 1.50

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

P0 = 1.99

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**¿En qué otro intervalo tiene la función f(x) otra raíz real?**

En (-2, -1)

## Encuentre una aproximación a una raíz de la función f(x) = x^3 + 4x^2 - 10 = 0, utilizando los siguientes parámetros: p0 = 0.5; TOLERANCIA: TOL = 10^-10; Cantidad máxima de iteraciones: IT = 100.

Primera transformación: **g(x) = √(10/(x+4))**

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Segunda transformación: **g(x) = √((10 – x^3)/4)**

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

#include <iostream>

#include <conio.h>

#include <stdio.h>

#include <cmath>

#include <string>

#include <math.h>

#include <iomanip>

//AUTOR: Abner Andres Perez Villatoro

int i, IT, respuesta;

double p0, p, TOL, FP, errAbs, errRel;

//Se evalua la funcion f(x). La segunda funcion esta marcada como comentario si en caso quiere ejecutarla

double f(double p){

return (pow(p,4) - 3\*(pow(p,2)) - 3); //f(x) = x^4 – 3x^2 – 3

//return (pow(p,3) + 4\*pow(p,2) - 10); //f(x) = x^3 + 4x^2 - 10

}

//Se evaluan las transformaciones. El resto de transformaciones están marcadas como comentarios si en caso quiere evaluar otra transformación.

double g(double p){

return (pow(3\*(pow(p,2)) + 3, 0.25)); // f(x) = x^4 – 3x^2 – 3 -> g(x) = (3x^2 + 3)^(1/4)

//return (sqrt(3\*pow(p,2)+3)/p); // f(x) = x^4 – 3x^2 – 3 -> g(x) = (3x^2 + 3)^(1/2)/x

//return (sqrt(10/(p+4))); // f(x) = x^3 + 4x^2 – 10 -> g(x) = (10/(x+4))^(1/2)

//return (sqrt((10-pow(p,3))/4)); // f(x) = x^3 + 4x^2 – 10 -> g(x) = ((10 - x^3)/4)^(1/2)

}

void ingresoDatos(){

std::cout << "\tMETODO DE PUNTO FIJO \nIngrese el valor inicial p0: ";

std::cin >> p0;

std::cout << "Ingrese la Tolerancia: ";

std::cin >> TOL;

std::cout << "Ingrese la cantidad maxima de iteraciones: ";

std::cin >> IT;

}

//Agrega y alinea los titulos de cada columna para la tabla que se va a mostrar

void cabecera(){

std::cout << std::left << std::setw(15) << "\nIteracion" //Se usa la funcion setw() y left para alinear los datos a la izquierda

<< std::left << std::setw(20) << "|p0"

<< std::left << std::setw(20) << "|p [g(p0)]"

<< std::left << std::setw(25) << "|f(p)"

<< std::left << std::setw(25) << "|Error Absoluto"

<< "|Error Relativo\n";

}

//Muestra los datos de cada iteracion en una nueva linea.

void tabla(){

std::cout << std::left << std::setw(15) << i

<< std::left << std::setw(20) << std::setprecision(15) << p0 //Se usa la funcion setprecision() para que los datos muestren una cantidad especifica de decimales

<< std::left << std::setw(20) << std::setprecision(15) << p

<< std::left << std::setw(25) << std::setprecision(15) << FP

<< std::left << std::setw(25) << std::setprecision(15) << errAbs

<< std::left << std::setw(25) << std::setprecision(15) << errRel << "\n"; //Se inserta una nueva linea para que muestre los datos de la siguiente iteracion

}

//Esta funcion aplica el Metodo del Punto Fijo

void algoritmoPuntoFijo(){

i = 1;

while (i <= IT){

p = g(p0); //Evalua la funcion g(x) usando p0

if (fabs(p - p0) < TOL){

FP = f(p); //Evalua p en la funcion original f(x)

tabla(); //Muestra en pantalla los datos de la iteracion

std::cout << "\n\tSOLUCION: p = " << p << " f(p) = " << FP;

break;

}

FP = f(p); //Evalua p en la funcion original f(x)

errAbs = fabs((p - p0)/p); //Calcula el error absoluto

errRel = fabs(p-p0); //Calcula el error relativo

tabla(); //Muestra en pantalla los datos de la iteracion

i++;

p0 = p; // El valor de g(p) se convierte en el siguiente valor inicial

}

if (i > IT){

std::cout << "El metodo fracaso";

}

}

int main() {

do {

ingresoDatos(); //Se le pide al usuario que ingrese los datos

cabecera(); //Muestra en pantalla los titulos de cada columna

algoritmoPuntoFijo(); //Evalua el algoritmo y va mostrando en pantalla los datos de cada iteracion

std::cout << "\nDesea realizar otra busqueda? \n 1) Si 0) No \nIngrese un numero: ";

std::cin >> respuesta;

system("cls"); //Limpia la consola

}while (respuesta != 0); //Mientras la respuesta no sea 0 que el programa siga ejecutandose

return 0;

}