

Practica 7

Esta practica consta de 6 ejercicios. Aquí escribiré los códigos fuente(en azul) y comentare los resultados obtenidos al ejecutarlos(en negro). Los archivos *.cpp ,los archivos *.h y los archivos *.exe están en el disquete adjunto.

Ejercicio 1;

Estudiad la integral y la derivada de la función $f(x)=x \exp (x-10)$ en el intervalo $[0,4]$ del siguiente modo:

a) b) Si utilizamos el programa xderint.cpp necesitamos llegar a un orden de extrapolación de Richardson $n = 4$ para obtener un error de la milésima parte del resultado de la integral.

$$\int f(x)dx = \int x \exp (x-10) dx = 7.481863502_{E-3} \pm 4.194566941_{E-7}$$

c) Esta integral se puede resolver analíticamente en el intervalo $[0,4]$, el resultado es;

$$\int f(x)dx = \int x \exp (x-10) dx = 7.4816_{E-3}$$

d) e) Ahora calcularemos a mano la derivada de la función $f(x)$ mediante el uso del método de extrapolación de Richardson en el punto $x_i = 0.5$.

Analíticamente encontramos que el valor de $f'(x)$ calculado en el punto $x_i = 0.5$ es;

$$f'(0.5) = 1.122777448_{E-4}$$

Tabla de extrapolación de Richardson;

Partimos de un $h = 0.01$

$$D(0,0) = d_0(h) = 1/2h [f(x+h)-f(x-h)] = 1.12277_{E-4}$$

Con el primer termino de la tabla de extrapolación de Richardson consigo un resultado optimo con un error en la estimación del orden 10^{-5} .

Ejercicio 2;

Primero dibujaremos la función $f(x) = (1+x^4)^{1/2}$ con el Gnuplot en el intervalo $[-10,10]$.

Ahora calcularemos el área encerrada bajo la curva y su error sencillamente integrando la función $f(x)$ en el intervalo $[-10,10]$ mediante una extrapolación de Richardson de orden $n = 5$ utilizando el programa xderint.cpp.

$$\text{Área bajo la curva} = \int f(x)dx = \int (1+x^4)^{1/2} dx = 669.0475528 \pm 0.000211383317 u^2$$

Repetimos el proceso anterior calculando el área bajo la curva utilizando un procedimiento parecido. Este caso aprovecharemos la paridad de la función $f(x)$ para calcular el área en la mitad del intervalo y hallar el área total encerrada en todo el intervalo multiplicando este resultado por dos.

$$\begin{aligned} \frac{1}{2} \text{Área bajo la curva} &= \int f(x)dx = \int (1+x^4)^{1/2} dx = 334.5196247 \pm 3.951134033_{E-6} u^2 \\ \text{Área bajo la curva} &= 2 (334.5196247 \pm 3.951134033_{E-6} u^2) = 669.0392494 \pm 7.902268066_{E-6} \end{aligned}$$

Comprobamos que el error obtenido por este procedimiento es mucho menor que el calculado mediante el procedimiento anterior.

Para obtener un error comparable con el intervalo de integración completo debemos calcular con el programa xderint.cpp una extrapolación de Richardson de orden $n = 6$ o incluso $n = 7$.

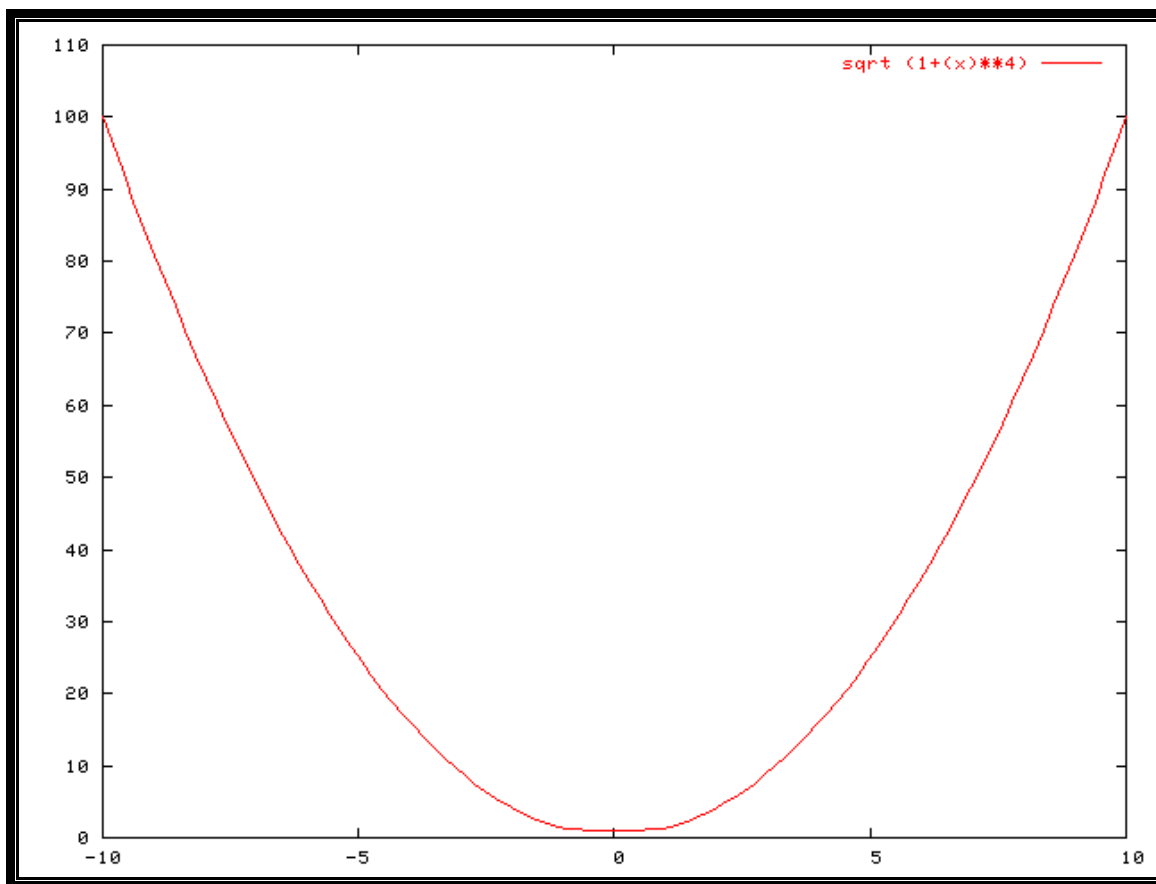


Fig 1; $f(x) = (1 + x^4)^{1/2}$ con el Gnuplot en el intervalo $[-10,10]$.

El error asociado al método de Richardson es menor cuanto menor es el intervalo de integración, por eso obtenemos un error mucho menor al integrar en el intervalo $[0,10]$ que al integrar en el intervalo $[-10,10]$. Esto se debe a que el error se calcula a partir de la diferencia entre los extremos del intervalo, cuanto mayor es la diferencia entre ellos mayor es el error cometido. Siempre que tengamos una función par podemos disminuir este error sencillamente calculando la integral en la mitad del intervalo y obteniendo el valor final de la integral multiplicando este resultado por dos, tal y como hemos actuado en el apartado anterior.

Ahora calcularemos el valor de la derivada de la función $f(x)$ mediante una extrapolación de Richardson de orden $n = 4$ en el punto $x_i = 0.5$. Analíticamente el resultado es;

$$f'(1) = 1.414213562$$

Utilizando el método de extrapolación de Richardson obtenemos que el resultado es;

$$f'(1) = 1.414213562 \pm 1.33226763_E-5$$

Ejercicio 3;

En este ejercicio programare una función que me permita calcular el valor de la integral definida en el intervalo $[a,b]$ de la función $f(x)$ por el método de la regla trapezoidal, el prototipo de la función será;

```
double trap( func F, double a, double b, int n);
```

donde F es la función a integrar, a y b los extremos del intervalo de integración y n es el numero de subintervalos en los que dividiréis el intervalo inicial. En realidad, en nuestro programa el numero de subintervalos total será el resultado de operar 2^n .

La regla trapezoidal permite calcular la integral de una función $f(x)$ en el intervalo $[a,b]$ de la siguiente forma;

$$\int_a^b f(x)dx \approx h/2[f(a) + f(b)] + h \sum_{i=1}^{2^n-1} f(a + ih)$$

sabiendo que $h = (b - a)/2^n$.

xp7(3).cpp

```
#include <iostream>
#include <iomanip>
#include <error.h>
#include <matutil.h>
#include <derint.h>

inline double F(double x)
{
    return x*exp(x-10.);
}

//Declaracion de la funcion
double trap(func F,double a,double b,int n);
//Definicion de la funcion
double trap(func F, double a, double b, int n){
    double h=(b-a)/(pow (2.,n));
    double fa=F(a);
    double fb=F(b);
    double l;
    double m=pow(2.,n)-1;

    for (int i=1;i<=m;i++){
        l +=F(a+i*h);
    }
    double j=0.5*h+h*l;
```

```

        return j;
    }

int main ()
{
    double a;
    double b;
    int n;

    cout << "Este programa calcula el valor de la integral definida en un intervalo [a,b] de
una funcion F" << endl;
    cout << "utlizando la regla trapezoidal recursiva para 2**n intervalos."<<endl;
    cout <<"Introduce el limite inferior a="<<endl;
    cin >> a;
    cout <<"Introduce el limite superior b="<<endl;
    cin >> b;
    cout <<"Introduce el numero n,el programa dividira el intervalo en 2**n subintervalos
iguales."<<endl;
    cin >> n;
    double Inte = trap(F, a, b, n);
    cout <<"El valor de la integral es I="<< Inte <<endl;

    return 0;
}

```

Compruebo el correcto funcionamiento del programa calculando la integral de la función $f(x) = x \exp(x - 10)$ en el intervalo $[0,4]$, obtenemos que;

$$\begin{aligned} \text{Para } n = 3, \quad ?f(x)dx &= ?x \exp(x - 10) dx = 2.0613 \\ \text{Para } n = 25 \quad ?f(x)dx &= ?x \exp(x - 10) dx = 0.00748215 \end{aligned}$$

El programa funciona correctamente. Comprobamos que el programa necesita muchos subintervalos para llegar al resultado correcto, a partir de un numero determinado de n el resultado de la integral es siempre el mismo.

Ejercicio 4;

En este ejercicio programare una función que me permita calcular el valor de la integral definida en el intervalo $[a,b]$ de la función $f(x)$ utilizando la regla de Simpson, el prototipo de la función será;

```
double simp( func F, double a, double b, int n);
```

donde F es la función a integrar, a y b los extremos del intervalo de integración y n es el numero de subintervalos en los que dividireis el intervalo inicial.

El procedimiento a seguir es el mismo que en el ejercicio anterior, solo que ahora el método de integración es distinto.

La regla de Simpson permite calcular la integral de una función $f(x)$ en el intervalo $[a,b]$ de la siguiente forma;

$$\int_a^b f(x)dx \approx \frac{h}{3} \sum_{i=1}^{n/2} [f(a+(2i-2)h) + 4f(a+(2i-1)h) + f(a+2hi)]$$

en este caso $h = (b - a)/n$.

Esta regla es valida para n pares, notemos que el sumatorio abarca el intervalo $i = 1$ hasta $n/2$, al definir n como entero en nuestro código es necesario que elijamos un n par para poder calcular correctamente el valor de la integral.

xp7(4).cpp

```
#include <iostream>
#include <iomanip>
#include <error.h>
#include <matutil.h>
#include <derint.h>

inline double F(double x)
{
    return x*exp(x - 10.);
}
//Declaracion de la funcion
double simp(func F, double a, double b, int n);
//Definicion de la funcion
double simp(func F, double a, double b, int n){
    double h=(b-a)/n;
    double m=n/2;
    double l;
    for (int i=1;i<=m;i++){
        l +=F(a+(2*i-2)*h)+4*F(a+(2*i-1)*h)+F(a+2*i*h);
    }
    double j=h*0.33*l;
    return j;
}
int main ()
{
    double a;
    double b;
    int n;
    cout <<"Este programa calcula el valor de la integral definida en un intervalo [a,b] de una
funcion F" << endl;
    cout <<"utilizando la regla de Simpson recursiva para n intervalos, donde n es
par."<<endl;
    cout <<"Introduce el limite inferior a="<<endl;
```

```

    cin >> a;
    cout <<"Introduce el limite superior b="<<endl;
    cin >> b;
    cout <<"Introduce el numero n(n debe ser un numero par!),el programa dividira el
intervalo"<<endl;
    cout <<"en n subintervalos iguales."<<endl;
    cin >> n;
    double Inte=simp(F, a, b, n);
    cout <<"El valor de la integral es I="<< Inte <<endl;
    return 0;
}

```

Compruebo el correcto funcionamiento del programa calculando la integral de la función $f(x) = x \exp(x - 10)$ en el intervalo $[0,4]$, obtenemos que;

$$\text{Para } n = 22, \quad \int f(x) dx = \int x \exp(x - 10) dx = 0.00740694$$

Como complemento el ejercicio 4 nos pide que escribamos las dos funciones en un mismo programa para, así, poder comparar resultados.

xp7(4a).cpp

```

#include <iostream>
#include <iomanip>
#include <error.h>
#include <matutil.h>
#include <derint.h>

inline double F(double x)
{
    return x*exp(x - 10.);
}
//Declaracion de la funcion
double simp(func F, double a, double b, int n);
double trap(func F,double a,double b,int n);
//Definicion de la funcion
double simp(func F, double a, double b, int n){
    double h=(b-a)/n;
    double m=n/2;
    double l;
    for (int i=1;i<=m;i++){
        l +=F(a+(2*i-2)*h)+4*F(a+(2*i-1)*h)+F(a+2*i*h);
    }
    double j=h*0.33*l;
    return j;
}
double trap(func F, double a, double b, int n){

```

```

double h=(b-a)/(pow (2.,n));
double fa=F(a);
double fb=F(b);
double l;
double m=pow(2.,n)-1;

for (int i=1;i<=m;i++){
l +=F(a+i*h);
}
double j=0.5*h+h*l;
return j;
}
int main ()
{
double a;
double b;
int n;
cout <<"Este programa calcula el valor de la integral de una funcion F(x) en un intrvalo
[a,b] utilizando"<<endl;
cout <<"el metodo de Simpson(para n intervalos donde n es par) y la regla trapezoidal
para 2**n intervalos."<<endl;
cout <<"Elegimos los limites del intervalo [a,b];"<<endl;
cout <<"Intrduce el limite inferior a="<<endl;
cin >> a;
cout <<"Introduce el limite superior b="<<endl;
cin >> b;
cout <<"Ahora elegiremos n(un n par!);"<<endl;
cin >>n;
double Intes=simp(F, a, b, n);
double Intet= trap(F, a, b, n);
cout <<"El valor de la integral utilizando el metodo de Simpson es I="<< Intes <<endl;
cout <<"El valor de la integral utilizando la regla trapezoidal recursiva es I="<< Intes
<<endl;
return 0;
}

```

Así, con un mismo n par podemos calcular el valor de la integral con dos métodos distintos. Utilizando el programa xderint.cpp comprobamos (para la función $f(x) = x \exp(x - 10)$) que el método de Simpson arroja la misma precisión que la regla trapezoidal con una extrapolación de Richardson. No se a que se debe.