

Practica 6

Esta practica consta de 5 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;

Primero generaremos con el programa xfundat.cpp un conjunto de 10 nodos igualmente espaciados de la función de Runge $(1 + x^2)^{-1}$ en el intervalo $[-5, 5]$. Después interpolaremos dichos nodos mediante un polinomio de grado 9. A continuación representaremos el resultado con el Gnuplot, superponiendo el valor real e la función y el resultado de la interpolación.

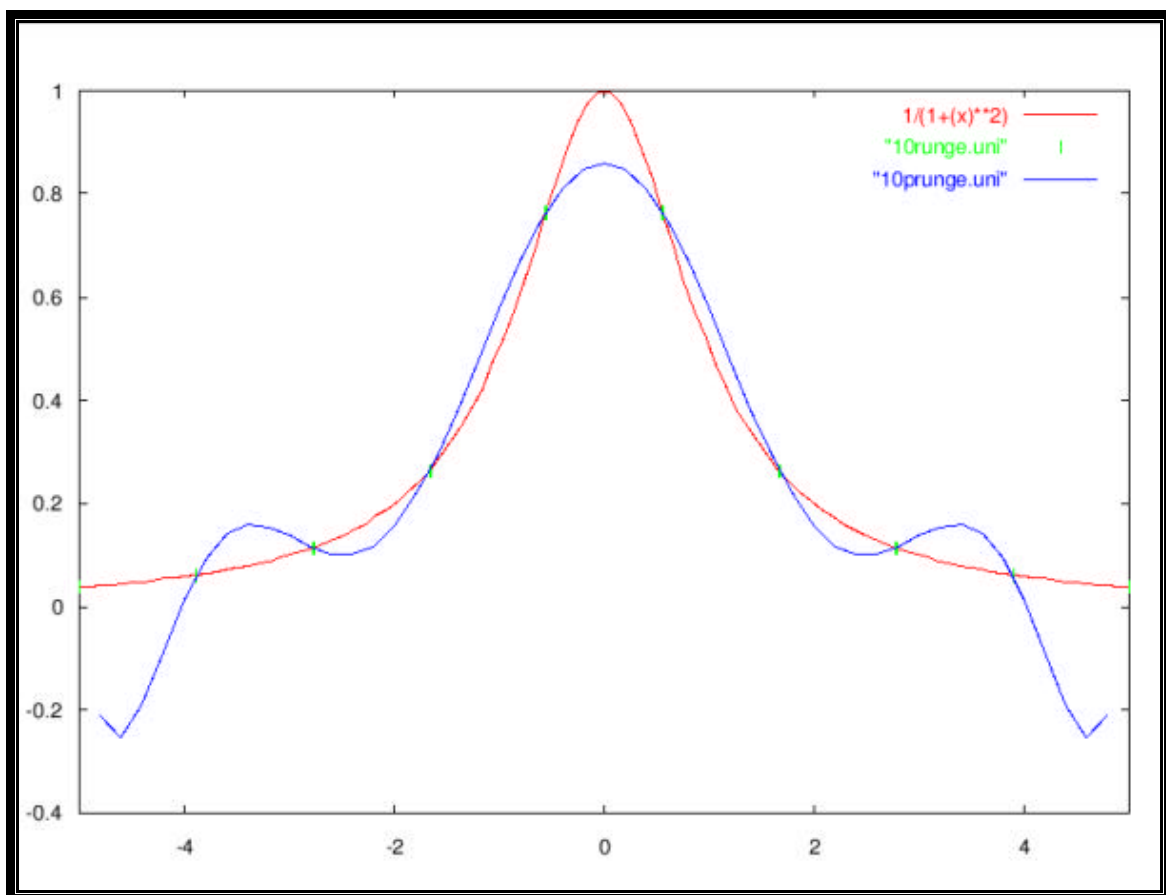


Fig 1; Función de Runge ajustada con 10 nodos igualmente espaciados.

La línea roja representa la grafica de la función de Runge. Los puntos verdes representan los 10 nodos igualmente espaciados obtenidos con el programa xfundat.cpp (guardados en el fichero "10runge.uni"), y la línea azul representa el polinomio de grado 9 ajustado a partir de esos nodos con el programa xinterpol.cpp (guardados en el archivo "10prunge.uni").

Ahora repetiremos el ejercicio generando 15 nodos. Obtenemos que;

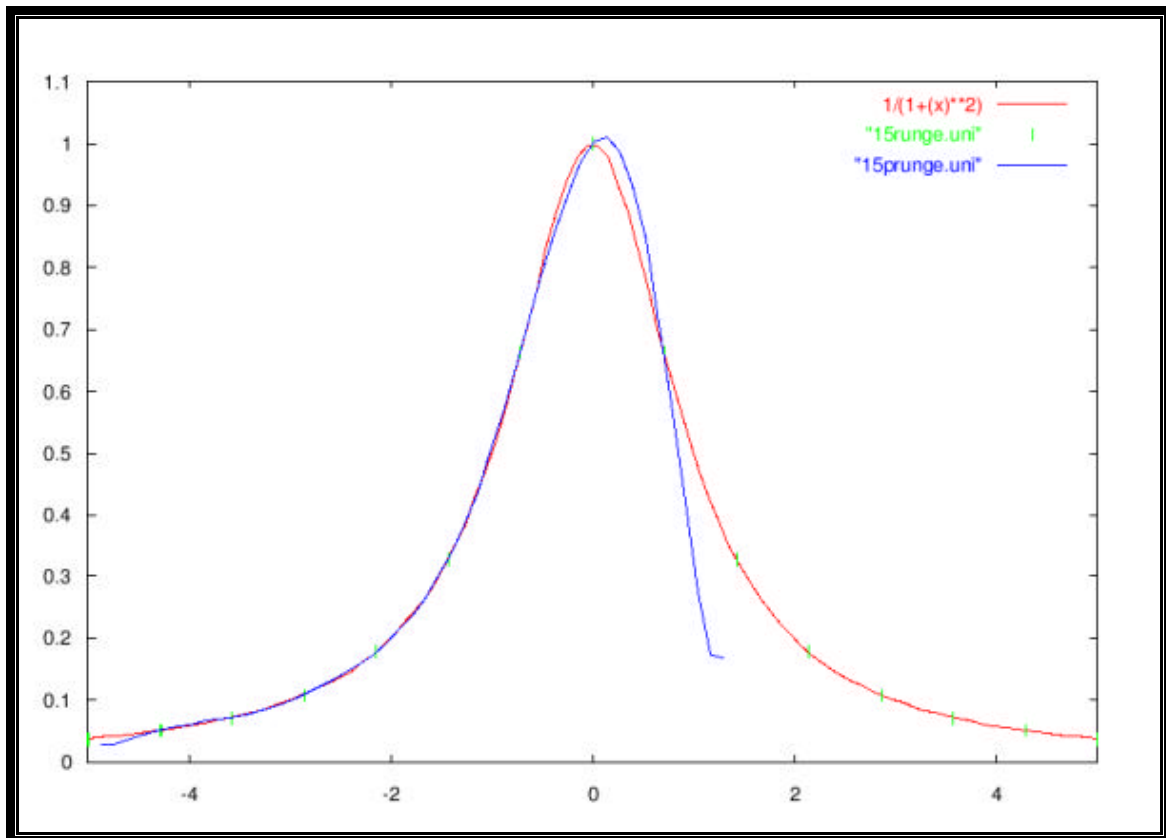


Fig 2; Función de Runge ajustada con 15 nodos igualmente espaciados.

La línea roja representa la grafica de la función de Runge. Los puntos verdes representan los 15 nodos igualmente espaciados obtenidos con el programa xfundat.cpp (guardados en el fichero “15runge.uni”), y la línea azul representa el polinomio de grado 9 ajustado a partir de esos nodos con el programa xinterpol.cpp (guardados en el archivo “15prunge.uni”).

Y ahora con 20 nodos. Obtenemos que;

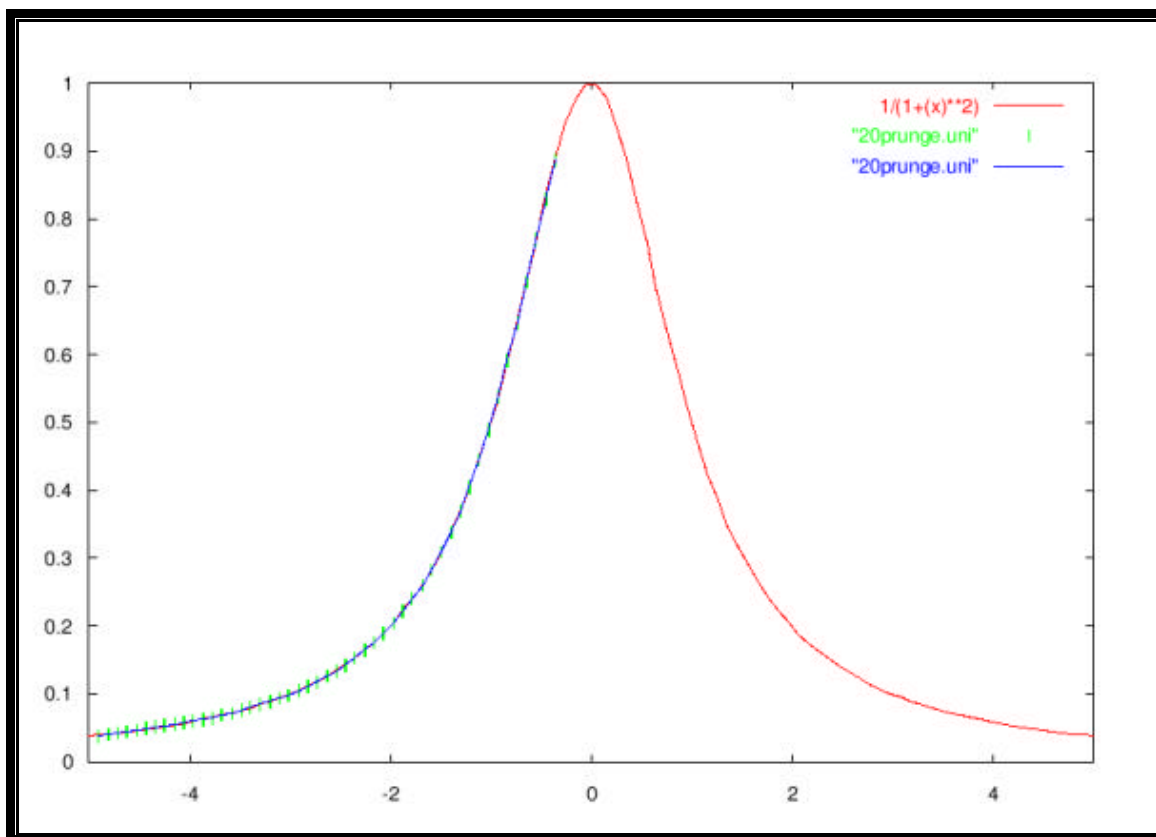


Fig 3; Función de Runge ajustada con 20 nodos igualmente espaciados.

La línea roja representa la grafica de la función de Runge. Los puntos verdes representan los 20 nodos igualmente espaciados obtenidos con el programa `xfundat.cpp` (guardados en el fichero “20runge.uni”), y la línea azul representa el polinomio de grado 9 ajustado a partir de esos nodos con el programa `xinterpol.cpp` (guardados en el archivo “20prunge.uni”).

A medida que aumentamos el numero de nodos la aproximación a los puntos intermedios de $f(x)$ es peor. El efecto de que la función $f(x)$ y el polinomio interpolador $P_n(x)$ coincidan cada vez en mas puntos es el de aumentar el error fuera de los nodos.

Este efecto se observa en las representaciones graficas anteriores. A medida que aumenta el numero de nodos el polinomio $P_n(x)$ se ajusta mejor a la función $f(x)$ pero solo en esos nodos. Al aumentar en numero de nodos, la aproximación del polinomio $P_n(x)$ en puntos intermedios empeora.

No siempre el argumento mas intuitivo es el mas correcto. Al aumentar en numero de nodos en un mismo intervalo disminuye la distancia entre ellos y el algoritmo interpolador tiene que ajustar un polinomio $P_n(x)$ para que sea coincidente con un mayor numero de puntos en un espacio mas reducido. Lo que estamos haciendo es constreñir la región de validez del polinomio interpolador $P_n(x)$. Así conseguimos una mayor aproximación de $P_n(x)$ a $f(x)$ en los nodos pero a costa de perder información de esa aproximación en los puntos intermedios. Por tanto, utilizar nodos igualmente espaciados no es siempre la elección mas adecuada.

Ejercicio 2;

Repetiremos el ejercicio anterior con una elección nodal mucho mas adecuada. Utilizando los llamados nodos de Chebyshev, que, en un intervalo $[a, b]$ se calculan como;

$$x_i = 0.5 * (a + b) + 0.5 * (b - a) \cos [(i/n) \pi] \quad (0 \leq i \leq n)$$

Para el caso de 10 nodos de Chebyshev obtenemos que;

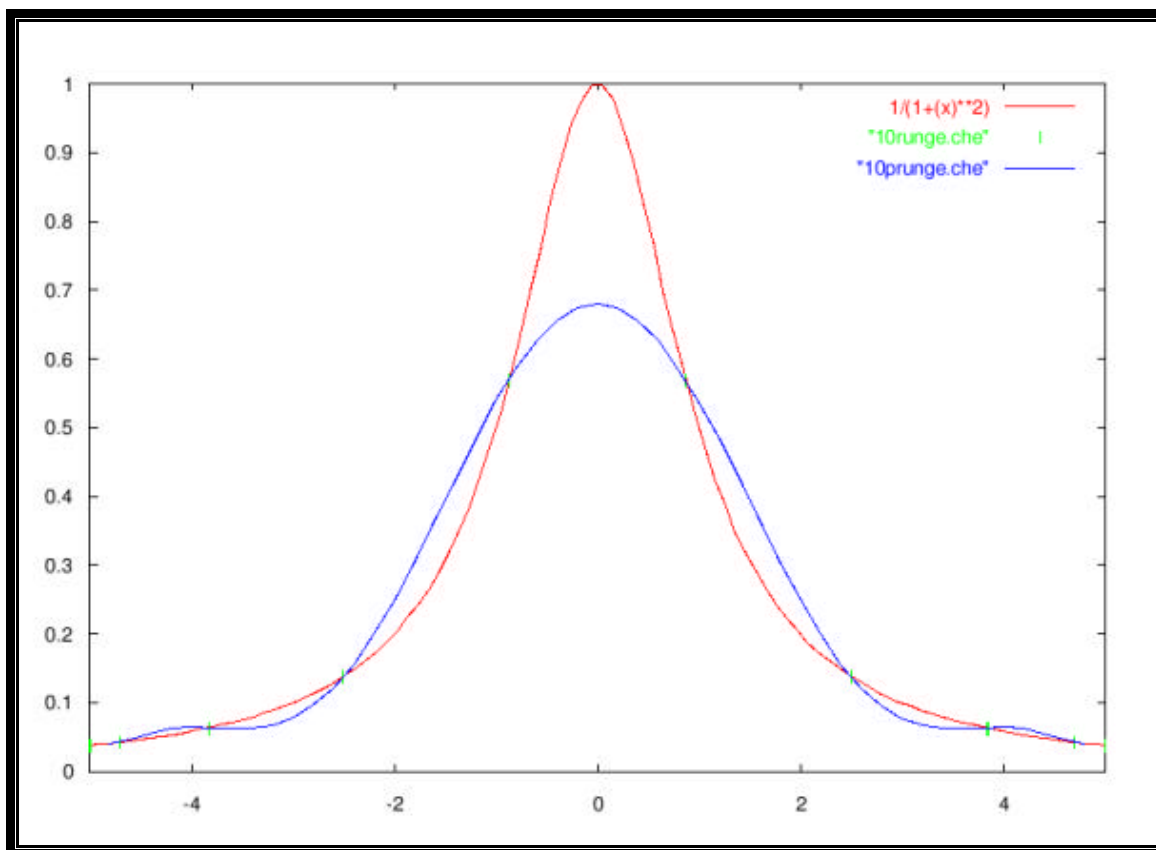


Fig 4; Función de Runge ajustada con 10 nodos de Chebyshev.

La línea roja representa la grafica de la función de Runge. Los puntos verdes representan los 10 nodos de Chebyshev obtenidos con el programa xfundat.cpp (guardados en el fichero “10runge.che”), y la línea azul representa el polinomio de grado 9 ajustado a partir de esos nodos con el programa xinterpol.cpp (guardados en el archivo “10prunge.che”).

A simple vista podemos observar que para el mismo numero de nodos (10 en este caso) el polinomio $P_n(x)$ calculado mediante los nodos de Chebyshev se aproxima mejor en los puntos intermedios de la función de Runge que en el ejercicio anterior.

Para el caso de 15 nodos de Chebyshev obtenemos que;

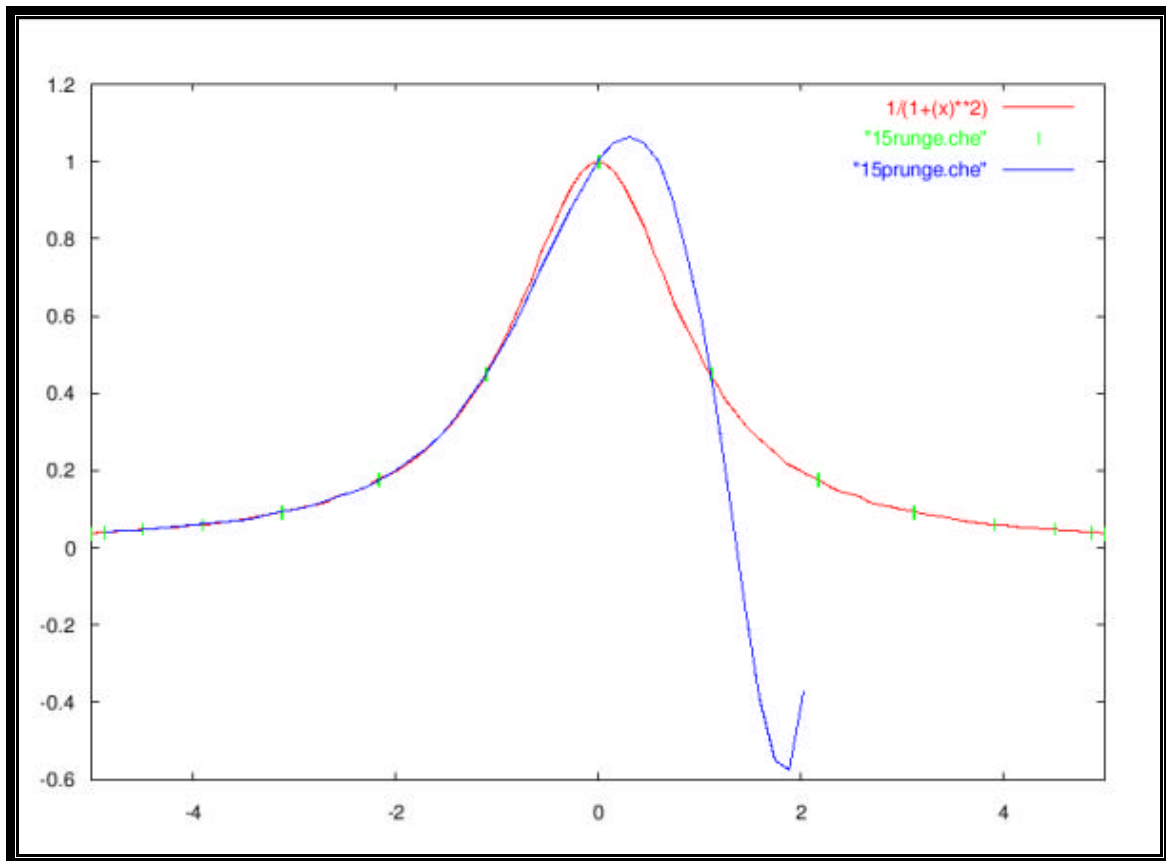


Fig 5; Función de Runge ajustada con 15 nodos de Chebyshev.

La línea roja representa la grafica de la función de Runge. Los puntos verdes representan los 15 nodos de Chebyshev obtenidos con el programa `xfundat.cpp` (guardados en el fichero “15runge.che”), y la línea azul representa el polinomio de grado 9 ajustado a partir de esos nodos con el programa `xinterpol.cpp` (guardados en el archivo “15prunge.che”).

Aquí observamos que al aumentar el numero de nodos la aproximación a la función $f(x)$ no mejora.

Para el caso de 20 nodos de Chebyshev obtenemos que;

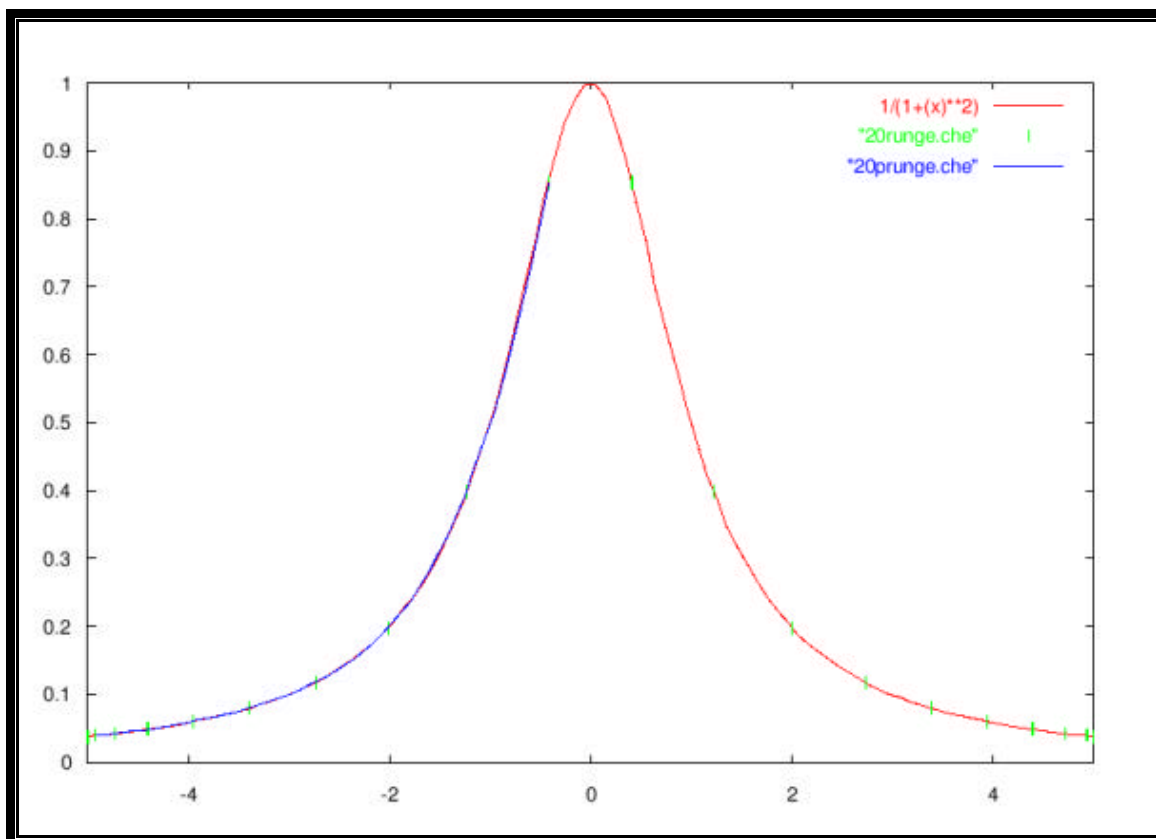


Fig 6; Función de Runge ajustada con 20 nodos de Chebyshev.

La línea roja representa la grafica de la función de Runge. Los puntos verdes representan los 15 nodos de Chebyshev obtenidos con el programa xfundat.cpp (guardados en el fichero “20runge.che”), y la línea azul representa el polinomio de grado 9 ajustado a partir de esos nodos con el programa xinterpol.cpp (guardados en el archivo “20prunge.che”).

Encontramos un resultado muy parecido al que obtenemos en el ejercicio anterior. La única diferencia es que utilizando los nodos de Chebyshev el Polinomio de interpolación $P_n(x)$ es mucho mas parecido a $f(x)$ en los puntos intermedios, lo que mejora el resultado final.

Por tanto, aunque la elección de los nodos de Chebyshev mejora el resultado final debemos fijarnos en que no solo la elección de nodos introduce errores en el calculo de polinomios interpoladores, además debemos fijarnos en el numero de nodos que elegimos a la hora de calcular ese polinomio.

Ejercicio 3;

En este ejercicio generaremos un conjunto de 10 nodos igualmente espaciados de la función $f(x) = \arctang(x)$ y buscaremos un polinomio de grado 9 que interpole a $f(x)$ en el intervalo $[-1, 1]$. Gráficamente obtenemos que;

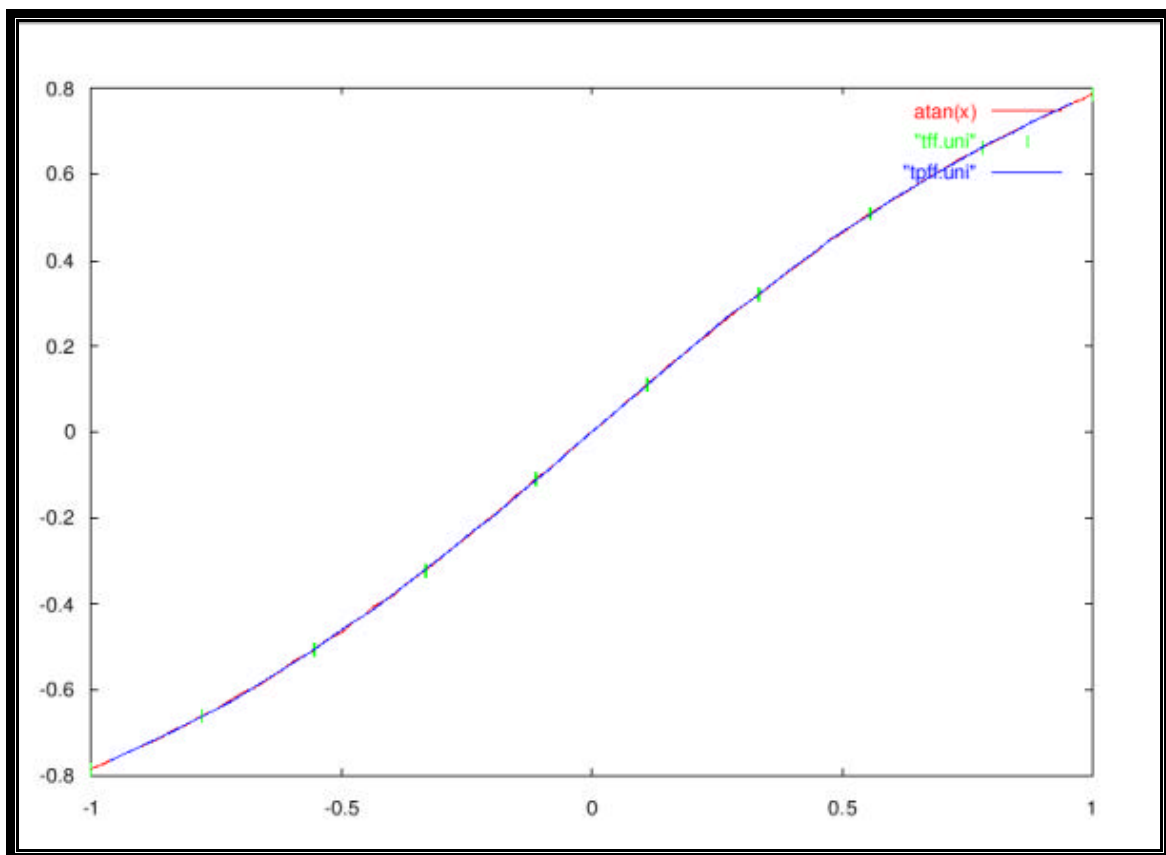


Fig 7; Función $f(x) = \arctan(x)$ ajustada con 10 nodos igualmente espaciados.

La línea roja representa la gráfica de la función $f(x) = \arctan(x)$. Los puntos verdes representan los 10 nodos igualmente espaciados obtenidos con el programa `xfundat.cpp` (guardados en el fichero "10tff.uni"), y la línea azul representa el polinomio de grado 9 ajustado a partir de esos nodos con el programa `xinterpol.cpp` (guardados en el archivo "10t9ff.uni").

Vemos que, pese a utilizar 10 nodos igualmente espaciados para calcular un polinomio $P_n(x)$ que interpole a la función $f(x)$, el resultado final es bastante bueno incluso en los puntos intermedios. Esto es debido a que $f(x)$ es casi lineal en el intervalo en el que hemos calculado la aproximación, por eso $P_n(x)$ se ajusta tan bien en los puntos internodales.

Ejercicio 4;

Ahora calcularemos el valor de la función $f(x) = \arctan(x)$ en el punto $x = -0.4$ con una precisión mayor que 10^{-4} .

Utilizaremos los programas `xinterpol2.cpp` y `xinterpol3.cpp`.

La única modificación que debemos introducir en este caso es definir la función de estudio. Por ejemplo, en el código de `xinterpol.cpp` deberíamos redefinir la función `ff` (como la función problema) de la siguiente manera;

```
inline double ff(double x)
{
```

```

    return atan(x);
}

```

El programa xinterpol2.cpp calcula hasta un polinomio de grado 7 donde;

$$|f(x) - P_n(x)| = 9.67602_{E-5}$$

El polinomio de grado menor que consigue una precisión mejor que 1_{E-4} es un polinomio de grado 5 donde;

$$|f(x) - P_n(x)| = 3.92038_{E-5}$$

Si no conociera la forma explícita de la función y me tuviera que guiar por la estimación del error a partir de las correcciones que aportan polinomios de grado consecutivo podría detener el proceso de añadir polinomios casi inmediatamente. Calculando un polinomio interpolador de grado 1 (una recta) obtengo que;

$$|f(x) - P_n(x)| = 6.9515431_{E-3}$$

Por tratarse de un error pequeño y porque la función $f(x)$ en este intervalo es casi lineal me conformaría con una recta.

Ahora utilizaremos el polinomio de interpolación que hemos calculado para estimar el valor de la función $f(x)$ en el punto $x_a = 1.5$.

El valor de la función $f(x) = \arctang(x)$ calculado en el punto x_a es $f(x_a) = 0.9827937232$.

El valor del polinomio de interpolación en ese mismo punto es $P_n(x_a) = 1.223698727$. Por tanto;

$$|f(x_a) - P_n(x_a)| = 0.2409050038$$

El resultado no es tan bueno como en el apartado anterior, esto se debe a que hemos calculado el polinomio de interpolación en un intervalo determinado y eso no asegura que fuera de ese intervalo la función $f(x)$ y el polinomio $P_n(x)$ converjan.

Ejercicio 5;

En este ejercicio generaremos un conjunto de 10 nodos igualmente espaciados de la función $f(x) = x \cdot \exp(x)$ y buscaremos un polinomio de grado 9 que interpole a $f(x)$ en el intervalo $[0, 2]$. Gráficamente obtenemos que;

La línea roja representa la gráfica de la función $f(x) = x \cdot \exp(x)$. Los puntos verdes representan los 10 nodos igualmente espaciados obtenidos con el programa xfundat.cpp (guardados en el fichero "10expff.uni"), y la línea azul representa el polinomio de grado 9 ajustado a partir de esos nodos con el programa xinterpol.cpp (guardados en el archivo "10expff.uni").

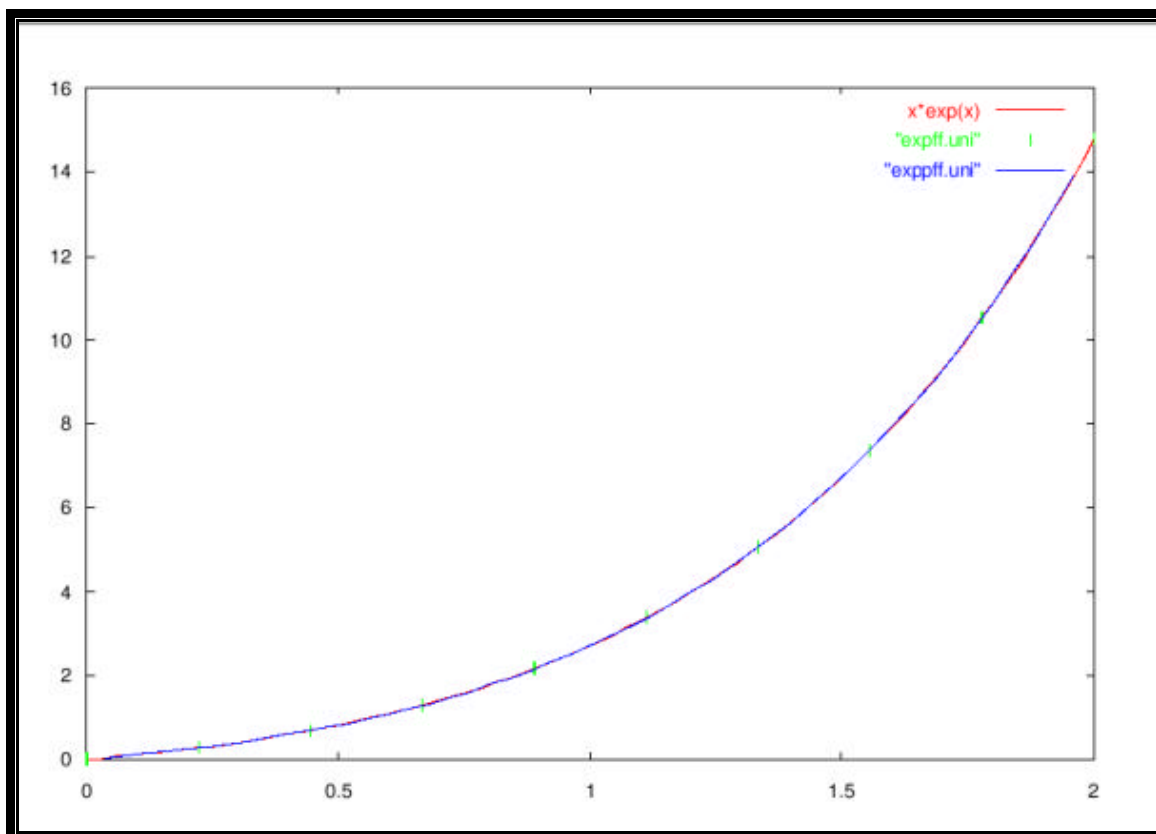


Fig 8; Función $f(x) = x \cdot \exp(x)$ ajustada con 10 nodos igualmente espaciados.

Aunque hemos utilizado nodos igualmente espaciados el polinomio interpolador de grado 9 $P_n(x)$ se ajusta bastante bien a $f(x)$ en este intervalo.

Por tanto podemos concluir que, en la mayoría de los casos, la elección de nodos igualmente espaciados no es incorrecta. La función de Runge es una excepción que nos ayuda a comprender la fragilidad del método con según que funciones. Para funciones como esta, o la función del ejercicio anterior, el resultado obtenido con los nodos igualmente espaciados y nodos de Chebyshev es prácticamente idéntica. Podemos comprobar la validez del método de equidistancia entre nodos haciendo una representación grafica como la que acabamos de realizar y, si vemos que el resultado no se ajusta en los puntos internodales(como pasaba en la función de Runge) repetir el calculo con los nodos de Chebyshev.

Ejercicio 6;

Ahora calcularemos el valor de la función $f(x) = x \cdot \exp(x)$ en el punto $x = 0.3$. Utilizaremos los programas xinterpol2.cpp y xinterpol3.cpp.

La máxima precisión que puedo encontrar la obtengo con un polinomio interpolador $P_n(x)$ de grado 6;

$$|f(x) - P_n(x)| = 3.1505 \cdot 10^{-6}$$

En el caso de que fuera yo el que interrumpe el proceso de adición de grados al polinomio interpolador me detendría en un polinomio interpolador de grado 4 donde;

$$|f(x) - P_n(x)| = 1.68337_E-4$$

Ahora utilizaremos el polinomio de interpolación que hemos calculado para estimar el valor de la función $f(x)$ en el punto $x_b = 3.0$.

El valor de la función $f(x) = x \cdot \exp(x)$ calculado en el punto x_b es $f(x_b) = 60.25661077$.

El valor del polinomio de interpolación en ese mismo punto es $P_n(x_b) = 57.44563579$. Por tanto;

$$|f(x_b) - P_n(x_b)| = 2.81097498$$

En este caso el error es mas pequeño pese a estar realizando una estimación fuera del intervalo donde hemos calculado el polinomio de interpolación $P_n(x)$. Eso es debido a que el comportamiento de la función $f(x)$ es muy parecido a lo largo de todo el grafo.

Después de estos resultados concluimos que para extrapolar resultados lo menos erróneos posibles es necesario estudiar con detalle el comportamiento de las funciones $f(x)$ a los que hacen referencia.