

Memoria de práctica 1, (Errores)
Métodos numéricos
 2n curso del grado en Matemáticas

Fecha de Reporte: **11/03/2019**
 Alumno: *Graells Ricardo, Marc*
 (NIU: 1388471)



N la práctica presente, tal como indica su título, se hace especial énfasis en los *errores*, muchas veces algo sutiles, inherentes al uso de la *aritmética en punto flotante* vista durante la primera semana del curso en la asignatura de *Métodos Numéricos*.

Recordando los conceptos fundamentales de programación en lenguaje C vistos en *Eines Informàtiques per a les Matemàtiques* del primer curso; adquirimos practica y experiencia en los errores típicos que surgen al intentar implementar un algoritmo *matemático* de forma finita mediante *computación*. Además y de forma más relevante intentamos entender el *porqué* de estos *errores* y en ultima instancia conocer estrategias para solventarlos. Aquí sigue un resumen de lo aprendido por puntos para futuras consultas.

- I A la hora de usar funciones en C hace falta saber con que precisiones trabaja cada una. Por ejemplo en el problema 1 y 2 se usan funciones `sinf()`, `sqrtf()` y `sin()`, `sqrt()`; para trabajar con valores `float` y `double` respectivamente.
- II Las variables `float` y `double` de IEEE tienen una precisión limitada (es decir, no podemos esperar mayor precisión de la teórica porque no es posible, la representación en punto flotante, esta sujeta a los *errores de representación*) relacionada con el concepto *épsilon* de máquina, que depende también de si la máquina *trunca* o *redondea*. Si suponemos el peor caso, la máquina redondea, podemos esperar errores relativos (entre el valor correcto y la aproximación numérica) por encima de 1.19×10^{-7} y 2.22×10^{-16} , respectivamente.
- III Los *errores de cancelación* parecen ser muy frecuentes y antes de obtener una buena implementación conviene sospechar y repasar que no sean presentes. Aunque con un uso mayor de memoria pueden resultar mucho menos dañinos.
- IV Con ejercicios 3, 4 y 5; fundamos la fe en que siempre parece haber una formulación más adie-te, y que en ocasiones una implementación mejor puede ahorrar el uso de variables con más memoria.
- V La aritmética en punto flotante no cumple la propiedad asociativa usual de las operaciones con números reales. Conviene recordar que cambiar el orden de algunos paréntesis o sumas puede mejorar notablemente el resultado obtenido. Ejercicios 3 y 5.
- VI No conviene subestimar la dificultad de un algoritmo, aparentemente fácil en un entorno *mate-mático ideal*, su implementación mediante métodos numéricos de forma plenamente satisfactoria puede ser algo muy laborioso. Un ejemplo de esto es el problema 5.
- VII Recordando los casos fatales de *Ariane 5*¹ en 1996 y *The Patriot Missile Failure*² en 1991. Nos invitan a ser cautos a la hora de asegurar la fiabilidad de una implementación y nos animan a ser minuciosos en la elaboración de estas.
- VIII Me ha hecho especial ilusión leer estas líneas de texto y las adjunto como motivación personal.

...
 En orden decreciente 1.644934066848226, difiere en $4.440892098500626 \times 10^{-16}$.
 Tiempo de calculo: 2.221285 minutos

¹<http://www-users.math.umn.edu/~arnold/disasters/ariane.html>

²<http://www-users.math.umn.edu/~arnold/disasters/patriot.html>