1. Aritmética finita. Análisis del error.

1.1. Cálculo numérico vs. cálculo simbólico.

La mayoría de los problemas matemáticos que surgen de sistemas reales de ingeniería no son manejables mediante cálculo simbólico. Los principales motivos son los siguientes:

1. No existe método simbólico:
   - Las integrales que aparecen en el cálculo de probabilidades:
   $$\int_0^1 e^{-x^2} dx$$
   - Cálculo de polos en un circuito resuelto mediante transformadas:
   $$s^6 + 3s^4 + 2s^3 + s + 1 = 0$$
2. El método simbólico es extremadamente lento:
   - Resolución de sistemas que surgen de redes con miles de nodos.
3. El método simbólico "inexplicablemente" falla al ejecutarlo en un ordenador
   - Para resolver la ecuación $x^2 + 10^{60}x + 1 = 0$ usamos la fórmula simbólica para la ecuación de segundo grado
   $$x = \frac{-10^{60} \pm \sqrt{\left(10^{60}\right)^2 - 4}}{2}$$
   Al simplificar con una calculadora o un ordenador nos queda
   $$x_1 = -9.9999 \times 10^{59},\ x_2 = 0$$
   ¡La segunda solución es falsa!
4. El modelo matemático es generalmente una aproximación de la situación real. Además, los datos de los que se dispone son habitualmente aproximaciones inexactas de los datos reales, así que no tiene sentido usar métodos exactos.

1.2. Tipos de métodos.

1. Métodos directos: Conseguimos mediante un número finito de operaciones la solución exacta (al menos en teoría, en la práctica no siempre es así). Se suelen trasladar directamente desde los métodos del cálculo simbólico. Algunos ejemplos son
   - Fórmula de la ecuación de segundo grado.
   - Método de Gauss para resolver sistemas.
   - Método de Barrow para integrales.
2. Métodos iterativos. Generamos una sucesión (x_n) tal que (al menos en teoría) $\lim_{n\to\infty} x_n = \hat{x}$, donde $\hat{x}$ es solución del problema. Por ahora entendemos límite en un sentido coloquial, ya que los elementos x pueden ser números, vectores, funciones, ...

1.3. Representación de números.

Cuando un número no tiene representación decimal exacta o no nos interese guardar todos los decimales, lo podemos redondear con el número de cifras significativas que queramos, siguiendo las mismas reglas que para redondear euros. En general para escribir un número decimal en notación científica escribimos

$$x = \pm 0.d \times 10^{\pm k}$$

Los ordenadores tienen una memoria limitada, así que esto lo tienen que hacer muy a menudo. Los números se guardan en base 2: sólo se guardan ceros (0) y unos (1). Muchos números que en decimal son exactos, en binario son periódicos. Por ejemplo, $0.1_{diez} = 0.00\widehat{0011}_{dos}$ es un número periódico. En los PCs ordinarios, para cada número se reservan generalmente 64 bits (espacios de memoria). Esto se llama representación en doble precisión. Así, cada número se codifica como

$$x = (-1)^s \times 1.m \times 2^{e-1023}$$

donde s se llama signo y ocupa 1 bit, m se llama mantisa y ocupa 52 bits y e se llama exponente y ocupa 11 bits.

Esto quiere decir que el ordenador tiene una precisión de 52 bits. Esta precisión define un número bastante importante: el épsilon de la máquina. En doble precisión es `eps`$= 2^{-52}$. Es la cantidad más pequeña comparable con el número 1. En un ordenador se tienen las siguientes igualdades:

$$1 + \texttt{eps} - 1 = \texttt{eps}$$

pero

$$1 + \frac{\texttt{eps}}{2} - 1 = 0$$

Este fenómeno se conoce como **error de redondeo**

¡Ojo! `eps` no es el número más pequeño representable en el ordenador. Los números más pequeño `realmin` y más grande `realmax` vienen definidos por las limitaciones en el tamaño del exponente. Al tener 11 bits, se tiene que (para números de la norma IEEE) $0 < e < 2047$ y por tanto $-1023 < e - 1023 < 1024$. Cualquier cálculo que de lugar a números mayores que `realmax` va a dar un error de **overflow**. En Matlab se representa como infinito `Inf` y no se paran los cálculos. Si por el contrario intentamos conseguir un número x tal que $0 < x <$ `realmin` dará un error de **underflow**. En general se representa como $x = 0$ y se siguen los cálculos.

1.4. Midiendo errores

Sea x una magnitud (escalar, vectorial, ...) y x_{ord} la magnitud que obtenemos en su lugar. Usemos el símbolo $\|x\|$ para medir esa magnitud (valor absoluto, norma de algún tipo, ...)

- Error absoluto:

$$\|x - x_{ord}\|$$

- Error relativo:
$$\frac{\|x - x_{ord}\|}{\|x\|}$$

1.5. Principales efectos del error de redondeo

Si sumamos 0.1 cien mil veces con un ordenador, no obtenemos 10000, sino que tenemos

```
>> s=0;for k=1:1e5, s=s+0.1; end, s
s =
    1.000000000001885e+004
>> s-1e4
ans =
    1.884836819954217e-008
```

Los error absoluto y relativo cometidos al sumar 0.1 cien mil veces son:

$$|s - 10^4| = 1.8848 \times 10^{-8}$$

$$\frac{|s - 10^4|}{|10^4|} = 1.8848 \times 10^{-12}$$

¡La octava cifra decimal ya es errónea! Tenemos que el error de redondeo se propaga. Este error se llama **error por acumulación**.

Más peligroso es el **error de cancelación**. Ocurre cuando se restan números muy próximos. Haciendo "bien" los cálculos (luego explicamos cómo), se tiene que

$$\sqrt{10^{15}+1} - \sqrt{10^{15}} = 1.5811 \times 10^{-8}$$

Pero

```
>> sqrt(1e15+1)-sqrt(1e15)
ans =
  1.8626e-008
```

Los error absoluto y relativo cometidos al calcular $\sqrt{10^{15}+1} - \sqrt{10^{15}}$ son

$$|1.8626 \times 10^{-8} - 1.5811 \times 10^{-8}| = 2.8151 \times 10^{-9}$$

$$\frac{|1.8626 \times 10^{-8} - 1.5811 \times 10^{-8}|}{|1.5811 \times 10^{-8}|} = 0.1780$$

¡Sólo hay 1 cifra correcta!

Este problema aparece en la fórmula de la ecuación de segundo grado y en el cálculo de derivadas mediante la definición.

Para evitarlo, hay que intentar transformar las operaciones para no hacer restas de ese tipo. La solución dependerá de cada caso. En nuestro ejemplo, hay que multiplicar y dividir por una expresión "conjugada": $\sqrt{10^{15}+1}+\sqrt{10^{15}}$.

Nota: También genera errores de redondeo dividir por números pequeños en valor absoluto, ya que esta operación genera números grandes que son más propensos a producir error de cancelación.

1.6. Principales fuentes de error en los cálculos.

1. Errores debidos al redondeo: Afectan a todos los métodos.
2. Errores debidos al método: No afectan a los métodos directos. Sólo a los iterativos.
3. Errores debidos a la falta de precisión en los datos. Afectan a todos los métodos.

1.7. Ejercicios Tema 1

Ejercicio 1.1 *Aproximando cada cálculo con 3 decimales, calcula*

$$s=\sum_{i=1}^{6}\frac{1}{3^i}$$

Compara con el resultado $\hat{s}=0.499314128943759$. *¿Qué efecto ha causado el error de redondeo?*

Ejercicio 1.2 *Sean* $f(x)$ *y su polinomio de Taylor de orden 2 centrado en* $x_0=0$

$$f(x)=\frac{e^x-1-x}{x^2} \quad \text{y} \quad p(x)=\frac{1}{2}+\frac{x}{6}+\frac{x^2}{24}$$

Calcular $f(0.01)$ *y* $p(0.01)$ *usando 6 sólo 6 decimales para todos los cálculo y calcular en cada caso el error absoluto y relativo cometido si el resultado correcto es* 0.50167084168057542.

¿Qué efecto ha causado el error de redondeo al calcular $f(0.01)$*?.*

Teniendo en cuenta que sólo podemos manejar 6 cifras, ¿se ha cometido error al tomar $f(0.01)=p(0.01)$*?*

Repetir el cálculo usando 16 cifras (necesitarás Matlab, las calculadoras no suelen tener más de 10 ó 12 cifras).

Ejercicio 1.3 *Encuentre en cada caso una fórmula equivalente que evite el error de cancelación*

- $\ln(x+1)-\ln(x)$ *para* $x>>1$.
- $\sqrt{x^2+1}-x$ *para* $x>>1$.
- $\cos^2(x)-\sin^2(x)$ *para* $x\approx\pi/4$