==

Práctica 2 - Algoritmos

Table of Contents

Aritmética de punto/coma flotante en MATLAB

Lectura recomendada: Floating Points: IEEE Standard Unifies Arithmetic Model

Lecturas complementarias:

1. Floating Point Arithmetic Before IEEE 754
2. A Glimpse into Floating-Point Accuracy
3. Floating Point Numbers

Toby Driscoll (2021). IEEE 754 binary representation (función al final del documento), MATLAB Central File Exchange.

El formato de memoria de un valor de punto flotante de doble precisión IEEE 754: Ilustrado en forma de byte, los bits del exponente y la fracción están distribuidos como:

[0][0000000 0000][0000 00000000 00000000 00000000 00000000 00000000 00000000]

En notación decimal, un número x es igual a:

```
fl(x) = (-1)^S * (1 + M/(2^52)) *  2^(E-1023),
```

excepto para valores especiales 0, Inf, NaN y números denormalizados (entre 0 y REALMIN).

```
suma =
    32

S =
'0'
E =
'10000000100'
M =
'0000000000000000000000000000000000000000000000000000'
```

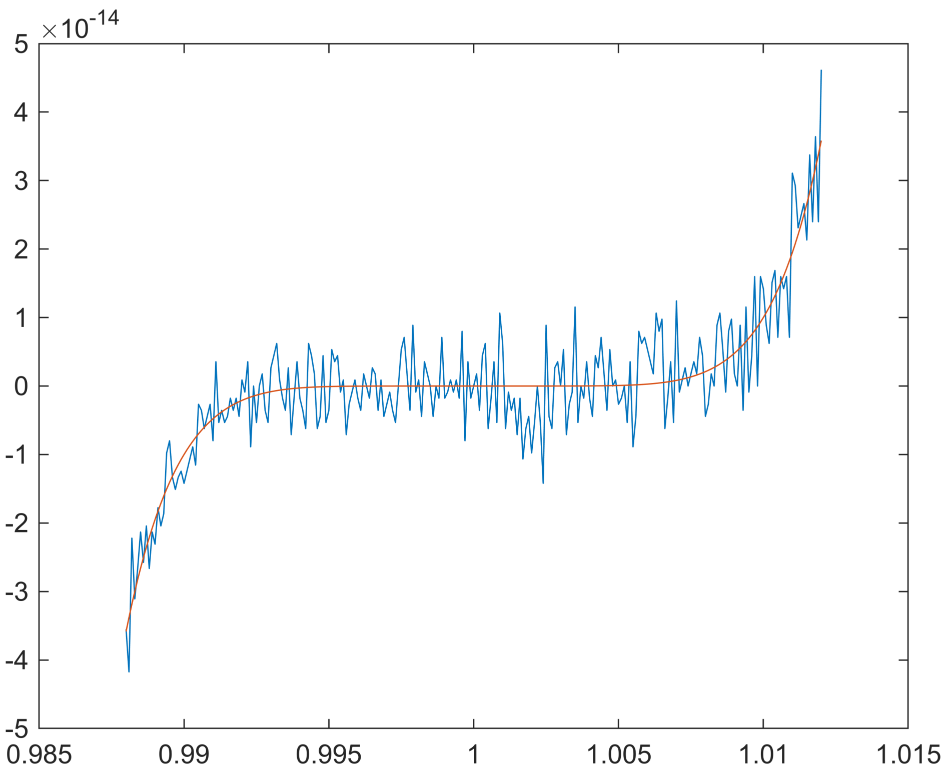
```
S =
     0
E =
        1028
M =
     0
```

Ejemplo 1. "Catastrophic Cancellation"

La cancelación catastrófica es el fenómeno por el cual restar buenas aproximaciones a dos números cercanos puede producir una muy mala aproximación a la diferencia de los números originales.

```
ans =
     0
```

Ejemplo 2. Ruido numérico



Ejercicios

Evitando problemas comunes con la aritmética de coma flotante

Ejercicio 1. Redondeo o lo que obtiene no es lo que se espera

- El número $a = 4/3$ no tiene mantisa binaria exacta, ya que todo número binario en coma flotante tiene una representación decimal exacta. Sea $b = a - 1$, calcular E (exponente) y M (mantisa) de la representación binaria en doble precisión de b, $2b$, y $3b$. ¿Por qué es esa la mantisa de $3b$? Comprobar que $c = 3b$ no es igual a 1. ¿Cuál es el error relativo? Compararlo con el épsilon de la máquina (eps en Matlab).

(Si se quieren ver las representaciones en binario puro se puede usar la función num2bin().BiCimal que está al final del documento.)

```
E_b =
'01111111101'
M_b =
'0101010101010101010101010101010101010101010101010100'

E_2b =
'01111111110'
M_2b =
'0101010101010101010101010101010101010101010101010100'

E_3b =
'01111111110'
M_3b =
'1111111111111111111111111111111111111111111111111110'

E_1 =
'01111111111'
M_1 =
'0000000000000000000000000000000000000000000000000000'

er =
     2.220446049250313e-16

E_er =
   971
M_er =
     0

E_er_unbiased =
   -52

ans =
     0
```

- Verificar que 0.1 no tiene representación binaria exacta. ¿MATLAB trunca o redondea la mantisa? Comprueba que sumar diez veces 0.1 no es igual a 1. Explicar qué está sucediendo.

```
M_01 =
'1001100110011001100110011001100110011001100110011010'

suma =
   1.000000000000000

ans = logical
   0
```

- Hay "gaps" entre los números en coma flotante: comprobar y argumentar que Matlab no distingue entre 2^{53} y $2^{53}+1$.

```
resta =
     0
```

```
menor_eps = logical
   1
```

- Comprobar que el seno de π no es cero.

```
seno_pi =
     1.224646799147353e-16

seno_pi_cero = logical
   0
```

Ejercicio 2. El orden de las operaciones puede hacer variar el resultado del cálculo

- Comprobar y argumentar que `1e-16 +1 - 1e-16` y `1e-16 - 1e-16 + 1` no dan el mismo resultado en MATLAB. ¿Cuál es correcto?

```
ans = logical
   0
```

- Comprobar y argumentar que `0.3 - 0.1 - 0.2 == -0.1 - 0.2 + 0.3` es falso en MATLAB.

```
ans = logical
   0
```

Ejercicio 3. Épsilon de la máquina

Determinar el épsilon de la máquina. Para ello, calcular $1 + x$ con $x = 2^{-i}$ para $i = 1, 2, \ldots$ mientras para MATLAB sea mayor a 1. Comparar con el `eps` de MATLAB.

```
     2.220446049250313e-16

     2.220446049250313e-16
```

Calcular los valores de $1 + \epsilon$ y $1 - \epsilon$. ¿Qué se observa? Argumentar el resultado.

```
ans =
     1

ans =
     9.999999999999999e-01
```

Ejercicio 4. Realmax

Hallar las potencias sucesivas de 10 con exponente desde 1 hasta que el valor de 10^k sea "infinito" para un exponente k y comprobar que entonces $\frac{1}{10^k}$ da 0. Comparar 10^k con realmax en doble precisión IEEE 754.

```
   309

   Inf

     0
```

Ejercicio 5. Números denormales

Calcular iterativamente la potencia de dos más pequeña distinta a cero en MATLAB. Compararla con realmin en doble precisión IEEE 754 y con eps*realmin.

Nota: la mayoría de los ordenadores permiten números denormales excepcionales entre realmin y eps*realmin (ver Floating Points: IEEE Standard Unifies Arithmetic Model).

```
       -1074

potencia_min =
    4.940656458412465e-324

eps_realmin =
    4.940656458412465e-324
```

Ejercicio 6. Propagación del error. Estabilidad numérica.

Para calcular las integrales $I_n = \int_0^1 x^n e^{x-1}dx\,, \quad n \geq 1$, disponemos de dos métodos iterativos diferentes:

a) $I_n = 1 - nI_{n-1}\,, \quad n \geq 2, \quad I_1 = 1/e.$

b) $I_{n-1} = \dfrac{1 - I_n}{n}, \quad n \geq 2, \quad I_{50} = 0.$

Discute la estabilidad de las recurrencias.

```
Ia = 1×50
     3.678794411714423e-01     2.642411176571153e-01     2.072766470286540e-01 ···

Ib = 1×50
     3.678794411714423e-01     2.642411176571153e-01     2.072766470286539e-01 ···

ind = 1×50
     1     2     3     4     5     6     7     8     9    10    11    12    13 ···
```

n	A	B
1.00000000000000e+00	3.67879441171442e-01	3.67879441171442e-01
2.00000000000000e+00	2.64241117657115e-01	2.64241117657115e-01
3.00000000000000e+00	2.07276647028654e-01	2.07276647028654e-01
4.00000000000000e+00	1.70893411885384e-01	1.70893411885384e-01
5.00000000000000e+00	1.45532940573080e-01	1.45532940573079e-01
6.00000000000000e+00	1.26802356561520e-01	1.26802356561528e-01
7.00000000000000e+00	1.12383504069363e-01	1.12383504069301e-01
8.00000000000000e+00	1.00931967445092e-01	1.00931967445593e-01
9.00000000000000e+00	9.16122929941707e-02	9.16122929896606e-02
1.00000000000000e+01	8.38770700582927e-02	8.38770701033942e-02
1.10000000000000e+01	7.73522293587803e-02	7.73522288626642e-02
1.20000000000000e+01	7.17732476946367e-02	7.17732536480296e-02
1.30000000000000e+01	6.69477799697233e-02	6.69477025756157e-02
1.40000000000000e+01	6.27310804238732e-02	6.27321639413801e-02
1.50000000000000e+01	5.90337936419019e-02	5.90175408792978e-02
1.60000000000000e+01	5.54593017295701e-02	5.57193459312356e-02
1.70000000000000e+01	5.71918705973076e-02	5.27711191689948e-02
1.80000000000000e+01	-2.94536707515363e-02	5.01198549580943e-02
1.90000000000000e+01	1.55961974427919e+00	4.77227557962091e-02
2.00000000000000e+01	-3.01923948855838e+01	4.55448840758181e-02
2.10000000000000e+01	6.35040292597259e+02	4.35574344078209e-02
2.20000000000000e+01	-1.39698864371397e+04	4.17364430279403e-02
2.30000000000000e+01	3.21308388054213e+05	4.00618103573729e-02

```
2.40000000000000e+01    -7.71140031330112e+06    3.85165514230504e-02
2.50000000000000e+01     1.92785008832528e+08    3.70862144237392e-02
2.60000000000000e+01    -5.01241022864573e+09    3.57584249827798e-02
2.70000000000000e+01     1.35335076174435e+11    3.45225254649453e-02
2.80000000000000e+01    -3.78938213288317e+12    3.33692869815312e-02
2.90000000000000e+01     1.09892081853613e+14    3.22906775355944e-02
3.00000000000000e+01    -3.29676245560839e+15    3.12796739321681e-02
3.10000000000000e+01     1.02199636123860e+17    3.03301081027895e-02
3.20000000000000e+01    -3.27038835596352e+18    2.94365407107357e-02
3.30000000000000e+01     1.07922815746796e+20    2.85941565457219e-02
3.40000000000000e+01    -3.66937573539107e+21    2.77986774454554e-02
3.50000000000000e+01     1.28428150738687e+23    2.70462894090608e-02
3.60000000000000e+01    -4.62341342659275e+24    2.63335812738122e-02
3.70000000000000e+01     1.71066296783932e+26    2.56574928689496e-02
3.80000000000000e+01    -6.50051927778940e+27    2.50152709799160e-02
3.90000000000000e+01     2.53520251833787e+29    2.44044317832742e-02
4.00000000000000e+01    -1.01408100733515e+31    2.38227286690335e-02
4.10000000000000e+01     4.15773213007410e+32    2.32681245696274e-02
4.20000000000000e+01    -1.74624749463112e+34    2.27387680756500e-02
4.30000000000000e+01     7.50886422691383e+35    2.22329727470511e-02
4.40000000000000e+01    -3.30390025984208e+37    2.17491991297495e-02
4.50000000000000e+01     1.48675511692894e+39    2.12860391612704e-02
4.60000000000000e+01    -6.83907353787311e+40    2.08421985815603e-02
4.70000000000000e+01     3.21436456280036e+42    2.04166666666667e-02
4.80000000000000e+01    -1.54289499014417e+44    2.00000000000000e-02
4.90000000000000e+01     7.56018545170645e+45    2.00000000000000e-02
5.00000000000000e+01    -3.78009272585323e+47    0.00000000000000e+00
```

Ejercicio 7. Propagación del error. Sistema lineal mal condicionado.

Resuelve el sistema $\begin{cases} 2x - 4y = 1 \\ -2.998x + 6.001y = 2 \end{cases}$ utilizando la instrucción linsolve de MATLAB.

Compara la solución con la del sistema $\begin{cases} 2x - 4y = 1 \\ -2.998x + 6y = 2 \end{cases}$.

¿Cómo son las dos soluciones? ¿Es un problema estable?

Calcula el número de condición de la matriz asociada al sistema lineal (funciones cond, rcond).

```
solA = 2×1
     1.400100000000000e+03
     6.998000000000001e+02
solQ = 2×1
     1.749999999999883e+03
     8.747499999999418e+02

condA =
     6.500000346149843e+03

rcondA =
     1.111123469273252e-04

condB =
     8.123500376894153e+03

rcondB =
     8.890864636586498e-05
```

Documento preparado por I. Parada el 21 de febrero de 2024