

Laboratorio 1: Sistema numérico de punto flotante

Sebastián Valencia Calderón
201111578

1 Introducción

La aritmética de punto flotante, proporciona las bases para la representación de cantidades reales en un computador digital. Dado que los cálculos numéricos son de fundamental importancia para el diseño de sistemas de ingeniería, y solución de grandes y complejos sistemas matemáticos, conviene entender los modos de representación de una cantidad real en un computador, y las implicaciones que esto tiene para los cálculos en cuestión. Las implicaciones, radican en la limitación básica del sistema; un computador digital, permite únicamente la representación de cierto conjunto de números con precisión finita. Esta limitación física para la representación de las cantidades, puede traer errores en los cálculos numéricos ejecutados por algoritmos implementados sobre un lenguaje de programación.

En este caso en particular, se estudia tal representación y sus limitaciones haciendo uso de MATLAB, una poderosa herramienta de computación científica, en la cual resulta pertinente la implementación de algoritmos numéricos, por su facilidad para representar y manipular arreglos de números. Mediante distintas aproximaciones, se pretende conocer las limitaciones de representación de la máquina o mejor, la arquitectura de la máquina sobre los números reales. A través de las implementaciones realizadas, se pretende cumplir con los siguientes objetivos:

- Entender las características, ventajas y desventajas asociadas con las unidades de punto flotante que permiten el desarrollo de la computación numérica.
- Observar casos que ilustran las precauciones que deben ser tenidas en cuenta en sistemas intensivos en computación numérica.

2 Procedimiento

Para cumplir los objetivos enumerados anteriormente, se desarrollan algoritmos para encontrar el epsilon (ϵ_M) de la máquina de un número particular. Con base en esto, se comparan los algoritmos, y de manera análoga, los resultados para un rango grande de los números representados en la máquina donde los algoritmos fueron ejecutados. Después, se procede con la experimentación sobre el rango de números representables, y las cantidades matemáticas fundamentales en MATLAB, posteriormente, se estudian y desarrollan algoritmos numéricos que manipulan la representación de los números para obtener las representaciones numéricas de los resultados ya conocidos de manera analítica.

De manera más desglosada, la ejecución de este laboratorio, procede de la siguiente forma:

1. Implementar un algoritmo numérico que halle de manera eficiente y certera el epsilon de la maquina (ϵ_M), esto es el valor que caracteriza el nivel de precisión de una máquina. El epsilon de la máquina, es el número de punto flotante más pequeño tal que: [1]

$$1 + \epsilon_M > 1$$

La definición dada anteriormente, es caracterizada únicamente por un número representado en punto flotante. Luego de esto, se pretende generalizar el epsilon sobre cualquier número, es decir, para un número x , $\epsilon(x) = \epsilon$; $|x + \epsilon| > x$. Con una distribución representativa de los números en punto flotante representables por la máquina de trabajo, se pretende comparar distintos algoritmos para su cálculo, y su valor en términos de x .

2. Con base en el estándar IEEE 754-2008 [2], se estudia la representación y de constantes matemáticas de importancia analítica, es decir, la representación numérica de algunos resultados analíticos fundamentales. De la misma manera, se calcula el valor de funciones matemáticas de interés y conocimiento general, esto, con el fin de dilucidar las implicaciones que pueda tener el desarrollo de modelos con base en cálculos numéricos sobre un computador digital.

3 Resultados

A continuación, se evidencian los resultados y la metodología y herramientas de ejecución para cada uno de los problemas propuestos en la guía de laboratorio.

1. "El valor epsilon de una maquina es la diferencia entre 1.0 y el numero de punto flotante siguiente. Ya que este espaciamento no es uniforme en todo el rango de representación de números de punto flotante, esta definición puede ser extendida alrededor de cualquier otro numero (diferente de 1.0) y escrita como $\epsilon_M(x)$ para cualquier numero de punto flotante x . Esta es una de las razones para calcular el valor de epsilon en MATLAB (el cual cuenta por defecto con un valor constante "eps"), adicionalmente, es conveniente conocer algunos algoritmos para computar este valor, ya que algunos lenguajes de programación no cuentan un valor predefinido del mismo." (Tomado de la guía de laboratorio).

En el script 2, se evidencia el diseño y escritura de un algoritmo para calcular ϵ_M . El script, muestra el algoritmo propuesto en el enunciado. Después de su ejecución, el valor fue:

$$\epsilon_M = 2.220446049250313 \times 10^{-16}$$

La constante eps de MATLAB, es: $2.220446049250313 \times 10^{-16}$.

2. Para generalizar la definición de ϵ_M , sobre cualquier número, se introduce el concepto de $\epsilon_M(x)$, es decir, el número en punto flotante más pequeño para el cual resulta que sumarlo a x , se tiene un número mayor a x en la representación del computador. Es decir, $\epsilon(x) = \epsilon$; $|x + \epsilon| > x$. Para diseñar un algoritmo que compute esto, resulta

pertinente recurrir al script 2, en este, se cuenta con un valor de inicialización de ϵ , de manera que $1.0 + \epsilon > 1.0$. Para la generalización requerida, se debe tener un ϵ , tal que $x + \epsilon > x$, de manera que se doble el valor de ϵ , y se incremente la suma $x + \epsilon$ hasta que $x + \epsilon \leq x$, se ha encontrado el valor $2 \times \epsilon$. En el script 3, se evidencia el desarrollo de un algoritmo, que análogamente al enunciado anteriormente, reduce el valor de ϵ , hasta que $x + \epsilon \leq x$.

Script 1: Gráfica de x vs $\epsilon(x)$

```

1  % Plots the value of the machine epsilon around a number
2
3  x = 0:1000000;
4  x = x.^100;
5
6  y = MaqEps(x);
7
8  plot(x, y);
9  set(gcf, 'Position', [400 400 700 700]);
10 saveas(gcf, '../img/plot_epsilon.png');
11
12 eps_max = max(y);
13 eps_min = min(y);
14
15 range = eps_max - eps_min;

```

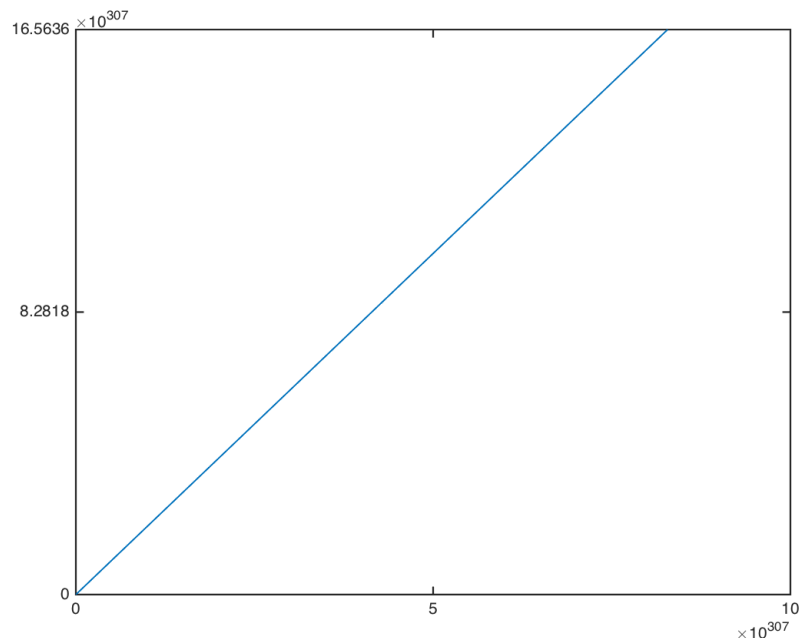


Figura 1: Un ejemplo

ANALISIS DE LA GRAFICA

3. El procedimiento diseñado por William Kahan, y utilizado en el paquete LINPACK, implementado en FORTRAN 77 [3], tiene como objetivo estimar las unidades de redondeo en una cantidad proporcional al tamaño de x . Es decir, el epsilon para un x en punto flotante. Sin embargo, asume ciertas características, que son riesgosas si no se usa este de manera adecuada. Por ejemplo, la base usada para la representación en punto flotante no debe ser potencia de tres. El código original, puede verse en el script 4.

Para analizar el desempeño de este y comparar sus resultados con los implementados previamente, se procede a implementar el algoritmo en MATLAB, realizar pequeños experimentos para comparar los valores, y posteriormente, realizar la prueba sobre un rango representativo de los números de punto flotante capaces de ser representados por la máquina digital que corre los programas.

4 Conclusiones

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

- Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.
- Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum. [?].

5 Bibliografía

- [1] Forsythe, G.E. and Malcolm, M.A. and Moler, C.B. *Computer methods for mathematical computations*, Prentice-Hall series in automatic computation. 1977.
- [2] IEEE Task P754. *IEEE 754-2008, Standard for Floating-Point Arithmetic*, pub-IEEE-STD. 2008.
- [3] Dongarra, J.J. and Bunch, J.R. and Moler, C.B. and Stewart, G.W. *LINPACK Users' Guide*, Society for Industrial and Applied Mathematics. 1979.

6 Scripts

Script 2: Cálculo de ϵ_M

```
1  %{
2      This script, computes dynamically the value of the machine
3      epsilon, executed in MATLAB/Octave. For this purpose, we use
4      the fact that the floating point representation of 1.0 + eps,
5      must be bigger than one. Then we decrease the value of eps by
6      a factor of (1 / 2.0), to find the required eps.
7
8      eps = epsilon
9  %}
10
11  format long;
12
13  epsilon = 1.0;
14  fl = 1.0 + epsilon;
15
16  while fl > 1.0
17      epsilon = epsilon / 2.0;
18      fl = 1.0 + epsilon;
19  end;
20
21  % We find the proper value, since the condition
22  % at the while loop increase the value of epsilon
23
24  epsilon = abs(2.0 * epsilon);
25  disp(epsilon);
```

Script 3: Cálculo de ϵ_M para un x específico

```
1  function [eps] = MaqEps(x)
2
3      eps = x;
4      fl = x + eps;
5
6      while fl > x
7          eps = eps / 2.0;
8          fl = x + eps;
9      end;
10
11      eps = abs(2.0 * eps);
12  end
```

Script 4: Cálculo de $\epsilon_M(x)$ según el algoritmo de Kahan implementado en FORTRAN

```

1  REAL FUNCTION EPSLON(X)
2      REAL X
3      REAL a , b , c , eps
4
5      a = 4.0/3.0
6      DO WHILE ( .TRUE. )
7          b = a - 1.0
8          c = b + b + b
9          eps = ABS(c-1.0)
10         IF ( eps.NE.0.0 ) GOTO 1
11     ENDDO
12 1   EPSLON = eps*ABS(X)
13     CONTINUE
14     END

```

Lista de Scripts

1	Gráfica de x vs $\epsilon(x)$	3
2	Cálculo de ϵ_M	6
3	Cálculo de ϵ_M para un x específico	6
4	Cálculo de $\epsilon_M(x)$ según el algoritmo de Kahan implementado en FOR- TRAN 77	6