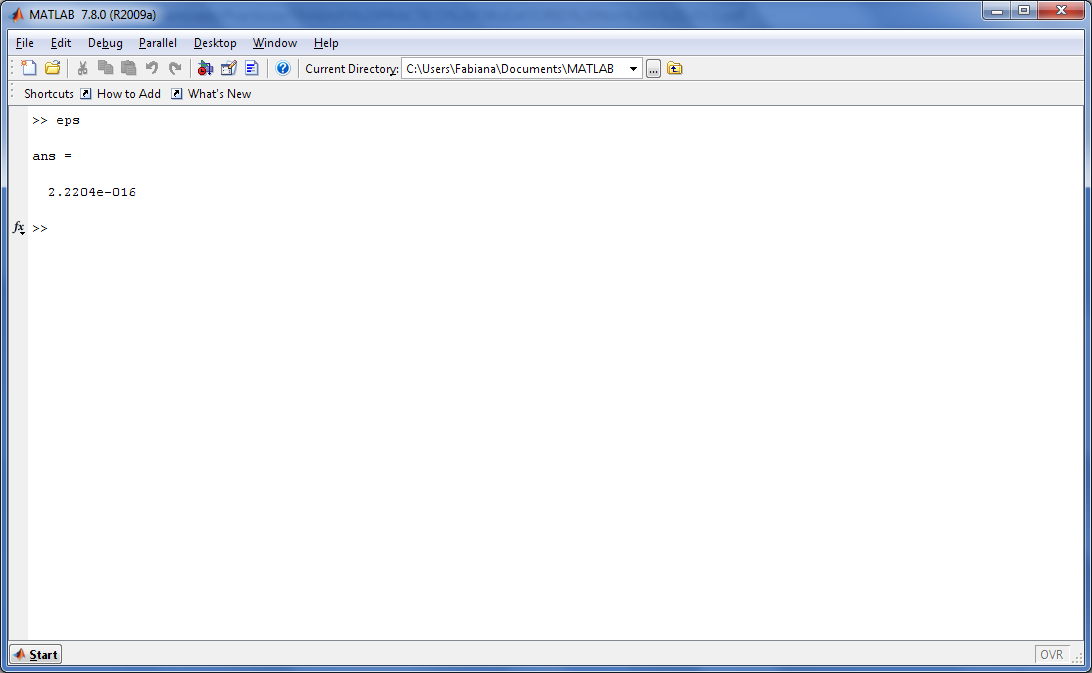
|  |
| --- |
| Métodos de Computación Científica - 2014 - |
| **Trabajo Práctico N° 1: “Introducción – Errores”**  **GARAT, Fabiana Yamel - LU 89108 -** |
|  |
|  |



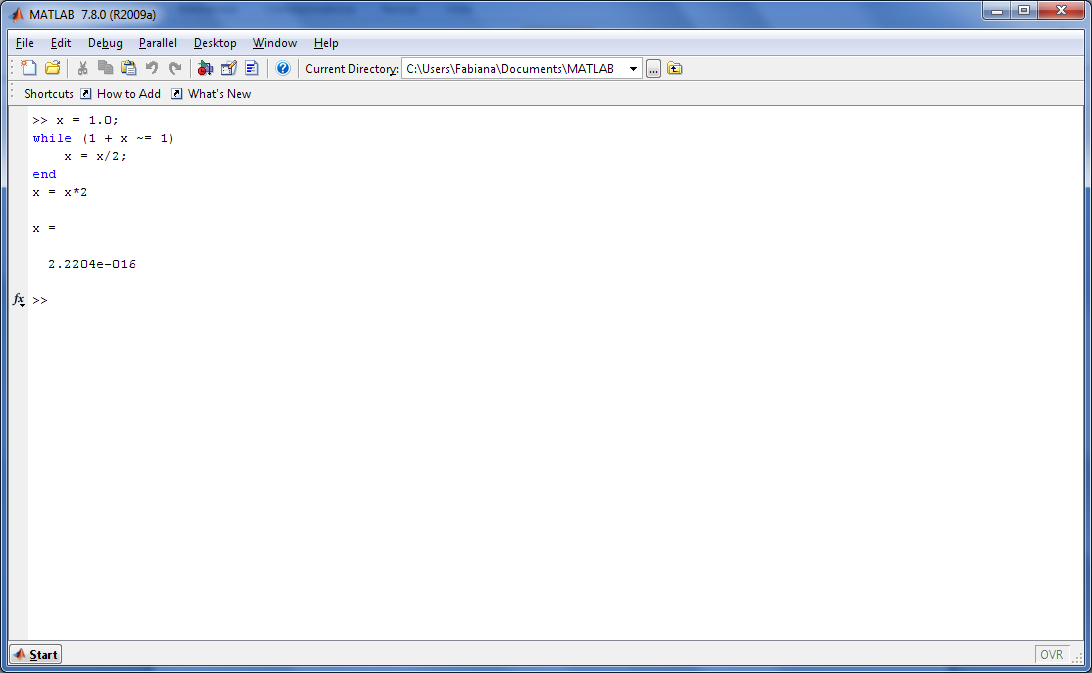
**Ejercicio 1 -** Encuentre el Epsilon machine de su computadora trabajando con Matlab.

**Solución –** Podemos obtener la solución a este problema de dos maneras diferentes. La primera opción es haciendo uso de la constante “eps” predefinida en Matlab de la siguiente manera:



**>> eps  
ans=  
 2.2204e-016**

La segunda opción es mediante el algoritmo:



**>> x = 1.0;**

**while (1 + x ~= 1)**

**x = x/2;**

**end**

**x = x\*2**

**x =**

**2.2204e-016**

Como se ve el algoritmo consiste en dividir sucesivamente por dos al número uno, hasta que ese resultado incrementado en uno sea igual a uno. Luego el número encontrado en la última iteración es multiplica por dos para obtener al épsilon machine (pues el número más chico que no hace que x+1 sea igual a 1 es el de la iteración anterior).

**Ejercicio 2 -** La constante de amortiguación *c* de la suspensión mostrada en la figura está dada por:



Donde µ es la viscosidad del fluido.

Sugiera un procedimiento para encontrar la influencia de pequeños errores en *a*, *h*, *r* y *l* sobre *c* para los valores de referencia: µ=0.3445 Pa.s, *l*=10 cm, *h*=0.1 cm, *a*=2cm, y *r*=0.5 cm.

Utilice el procedimiento para predecir el valor de *c* bajo las siguientes condiciones:

1. *l*=9.999 cm, *h*=0.09 cm, *a*=1.999 cm y *r*=0.499 cm.
2. *l*=10.001 cm, *h*=0.101 cm, *a*=2.001 cm y *r*=0.501 cm.
3. *l*=9.999 cm, *h*=0.101 cm, *a*=2.001 cm y *r*=0.499 cm.
4. Extraiga conclusiones.

**Solución –** Para comenzar contamos con una función que calcula el valor de la constante de amortiguación según los valores de l, h, a y r. Haremos que el valor µ=0.3445 Pa.s está dentro de la función ya que no forma parte de las variables que tienen pequeños errores.

Definimos la función como:

function [c] = constanteDeAmortiguacion(l,h,a,r)

u = 0.3445;

c = ((6\*pi\*u\*l)/(h^3))\*(((a - h/2)^2)-(r^2))\*

(((a^2 - r^2)/(a - h/2)) - h);

end

Al evaluar la constante “c” con los valores de referencia obtendremos lo que sería el valor real de la función (evaluada en los valores reales de las variables):

>> l = 10; h = 0.1; a = 2; r = 0.5; c = constanteDeAmortiguacion(l,h,a,r)

c =

4.2056e+005

1. >> l=9.999; h=0.09; a=1.999; r=0.499; c=constanteDeAmortiguacion(l,h,a,r)

c =

5.8098e+005

**b)** >> l=10.001; h=0.101; a=2.001; r=0.501; c=constanteDeAmortiguacion(l,h,a,r)

c =

4.0835e+005

1. >> l=9.999; h=0.101; a=2.001; r=0.499; c=constanteDeAmortiguacion(l,h,a,r)

c =

4.0873e+005

**d)** Podemos observar que los errores en las variables de los últimos dos incisos no afectan en gran medida al resultado. Por otro lado, en el caso de a) se puede observar una diferencia mucho mayor como resultado de evaluar la constante con esos valores.  
En los 3 incisos, todas las variables tienen pequeños errores, pero en el primer inciso “h” vemos un error relativo muy grande:

Hablamos, entonces, de un error de entrada muy alto. Debido a esto se podríamos decir que la variable que más afecta al cálculo es “h”. Esto se debe a que dicha variable es la más cercana al 0, sumado a que la misma aparece elevada al cubo en la fórmula.

**Ejercicio 3 –** Usando una rutina para encontrar raíces, compute las 20 raíces computadas para del polinomio P donde



Use una rutina que sea capaz de computar raíces complejas en doble precisión la formula dada para P es la forma expandida del polinomio de Wilkinson

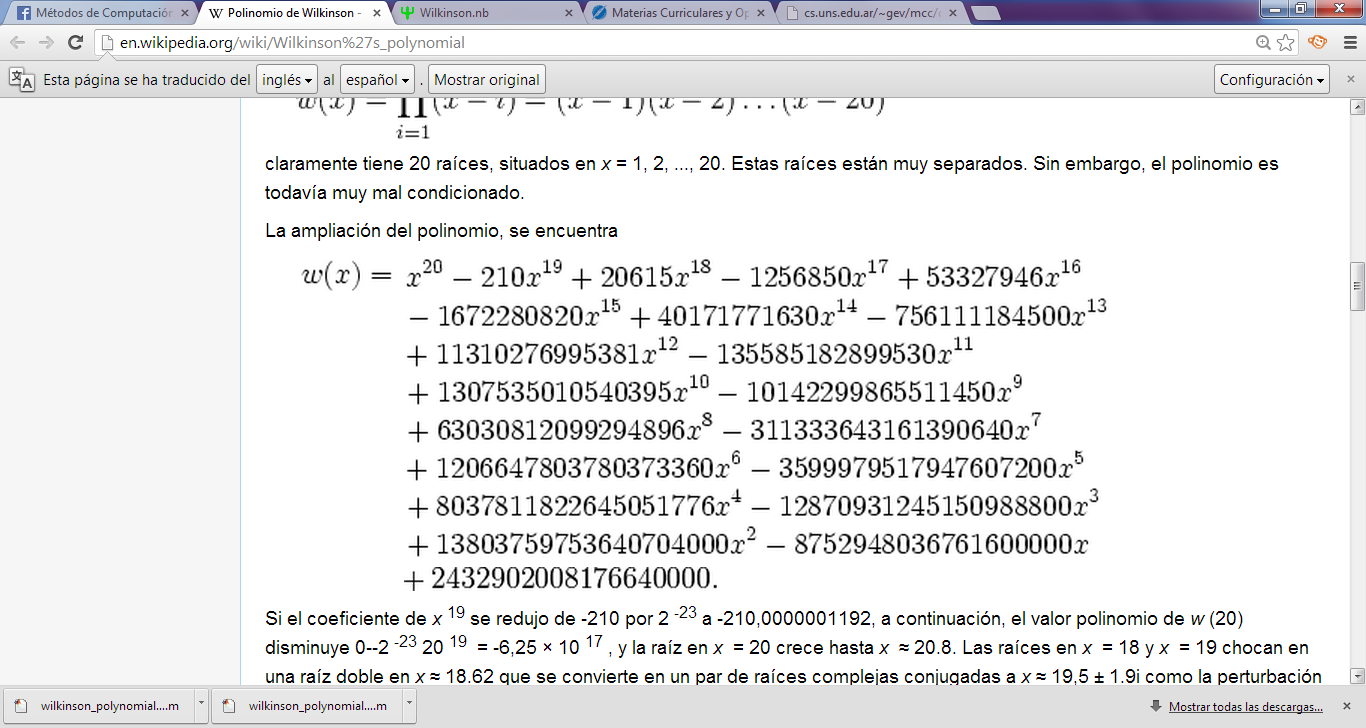
Cheque la calidad de las raíces computadas para calculando: . Explique

 w (x) = \ prod_ {i = 1} ^ {20} (x - i) = (x-1) (x-2) \ ldots (x-20). **Solución –** Se nos presenta el polinomio de la siguiente manera

Que es

Claramente tiene 20 raíces, situados en *x* = 1, 2, ..., 20. Estas raíces están muy separados. Sin embargo, el polinomio es todavía muy mal condicionado.

La ampliación del polinomio, se encuentra



Podemos calcular las raíces del Polinomio de Wilkinson mediante la siguiente rutina del entorno Matlab

>> p=[1 -1];

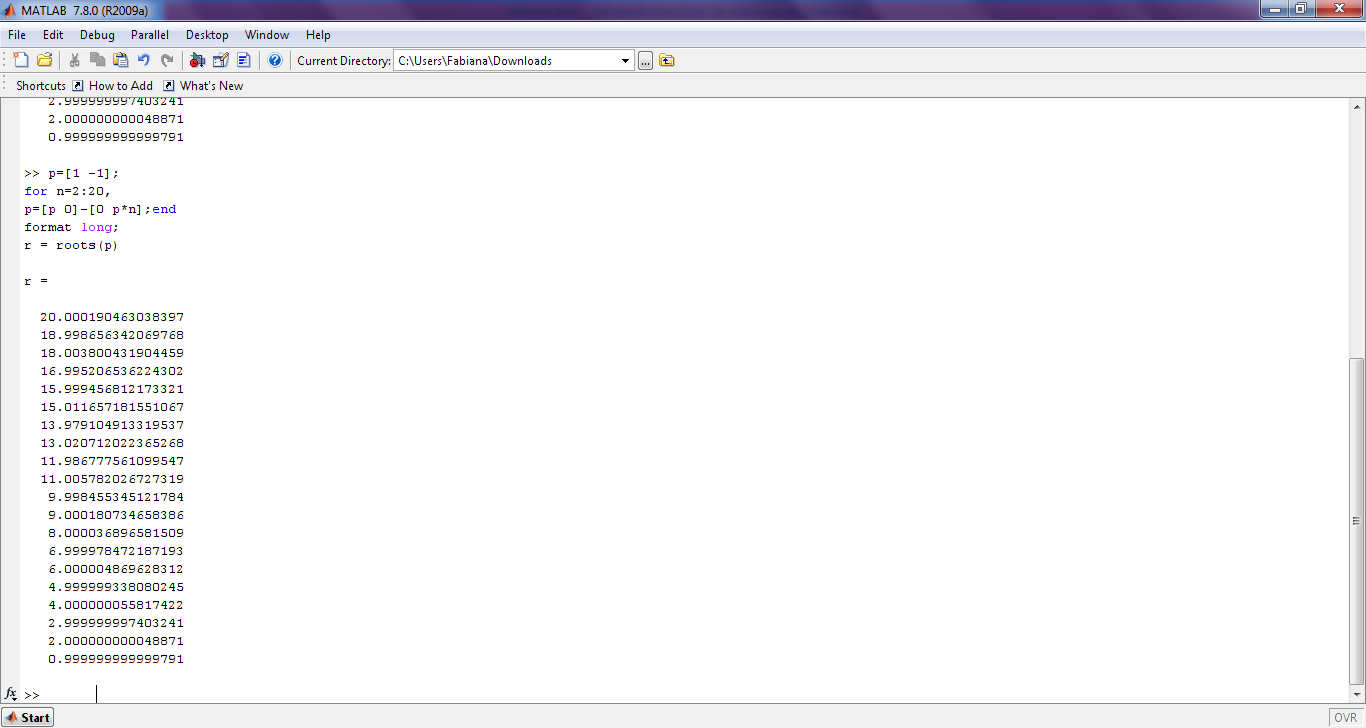
for n=2:20,

p=[p 0]-[0 p\*n];end

format long;

r = roots(p)

Retornando las raíces de la siguiente manera



Los valores de polinomios y sus raíces son extremadamente sensibles a la ronda de errores incluso si los cálculos se realizan con doble precisión.

Un cambio relativo en cualquiera de los coeficientes causa gran distorsión en las raíces. En algunos casos, incluso, llegamos a obtener raíces complejas. Pensemos en cambiar

La derivada de una raíz con respecto a *p*19  puede considerarse como una medida de la sensibilidad

Donde se podrá observar que los valores absolutos de exceden los , lo que explica las distorsiones de las raíces.