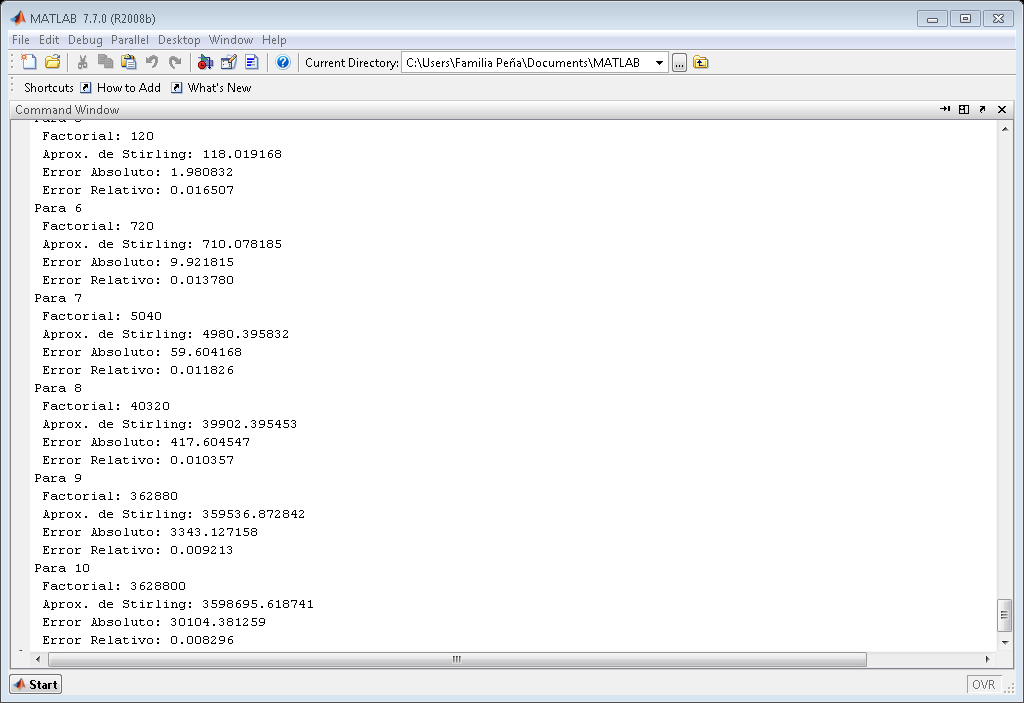
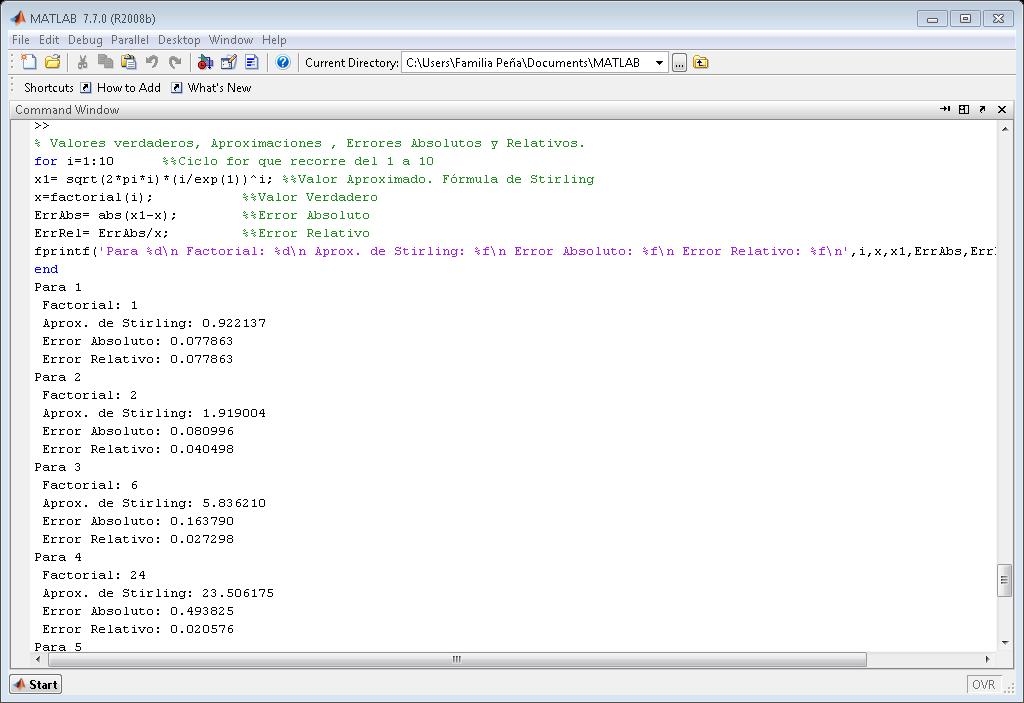
Peña Alarcón Raquel

ANÁLISIS NUMÉRICO

Aritmética del punto flotante y teoría de errores

1. Escribir un programa para calcular los errores absolutos y relativos en la aproximación de Stirling El error absoluto crece o decrece cuando n se incrementa? El error relativo crece o decrece cuando n aumenta? ¿Cuál es mejor?

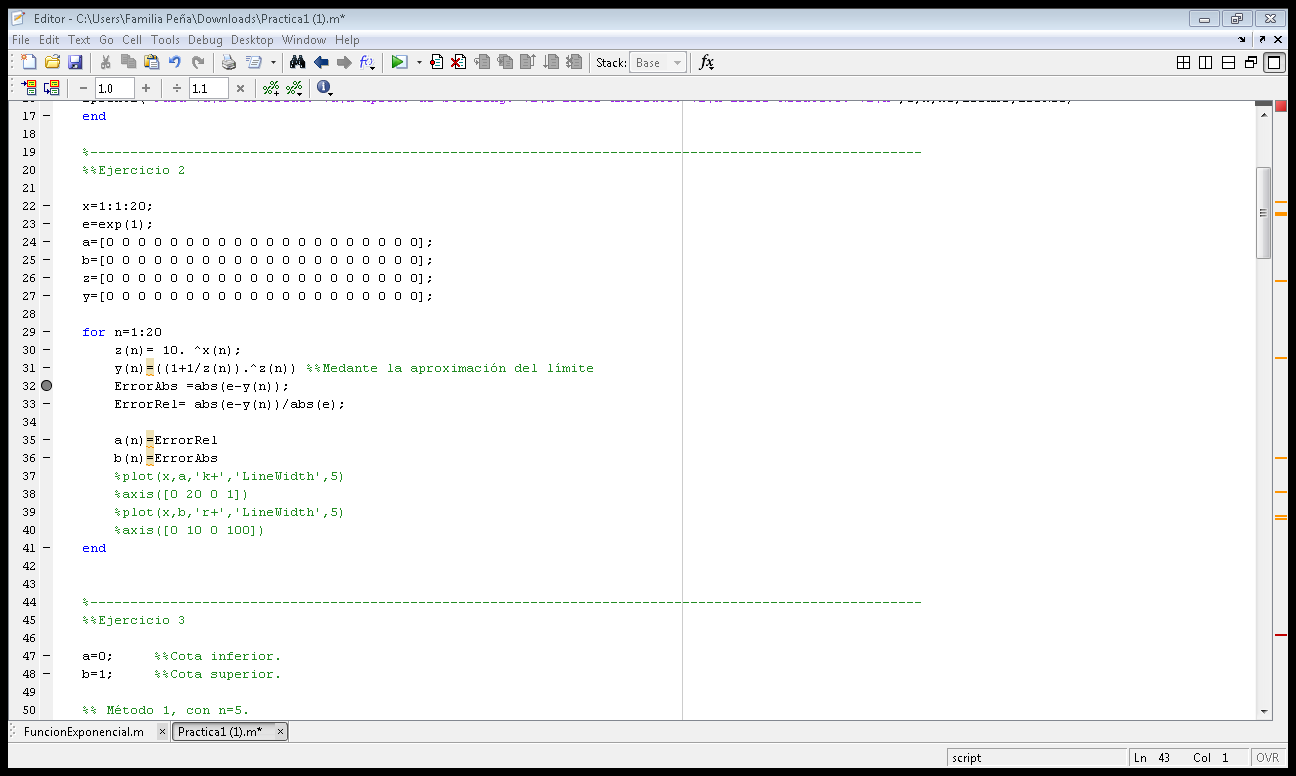


Se observa que el error absoluto y relativo son iguales para el primer número, n=1, sin embargo, conforme n aumenta, el error absoluto va incrementando, mientras el error relativo decrece acercándose a cero, recordando que el error relativo es más conveniente para trabajar pues está relacionado con la mantisa y la precisión.

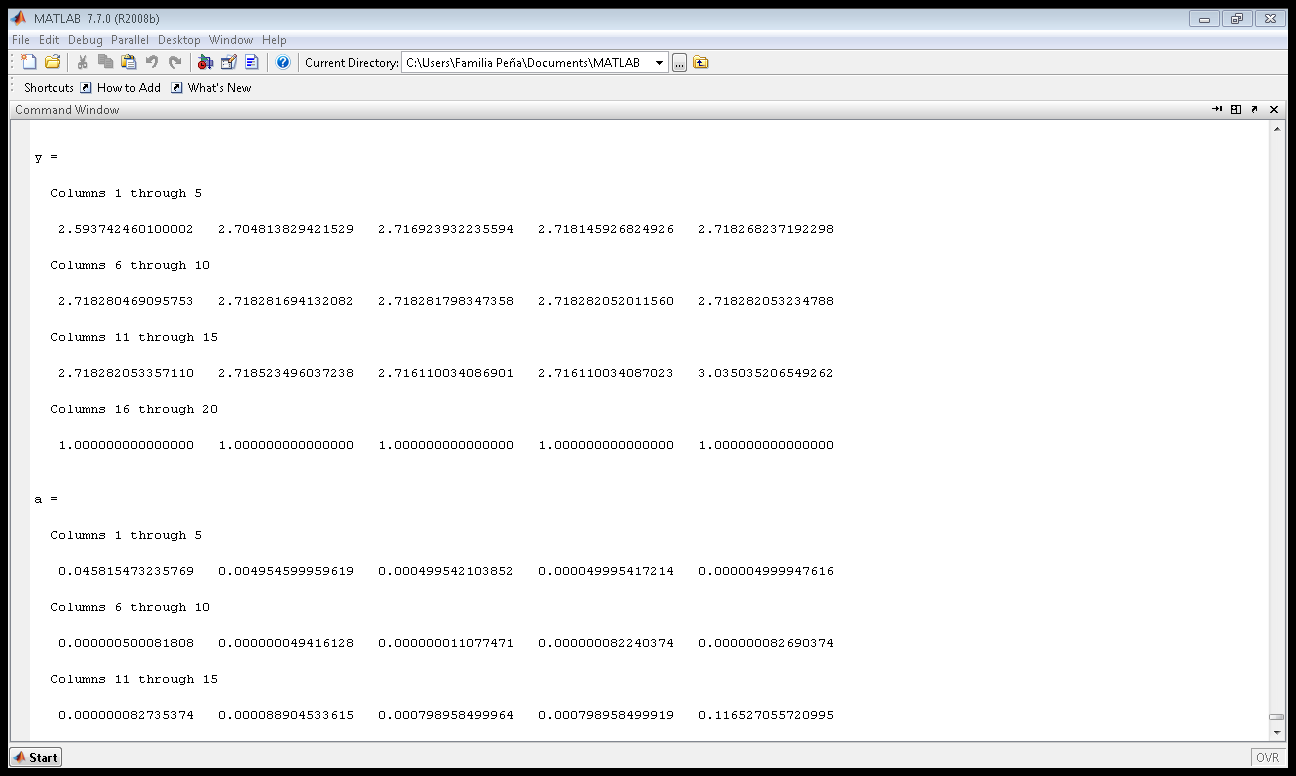
Entonces podemos concluir que la aproximación de Stirling es bastante buena para números grandes, en este caso, llegamos hasta 10! .

2. Escribir un programa para calcular la constante matemática e de la definición

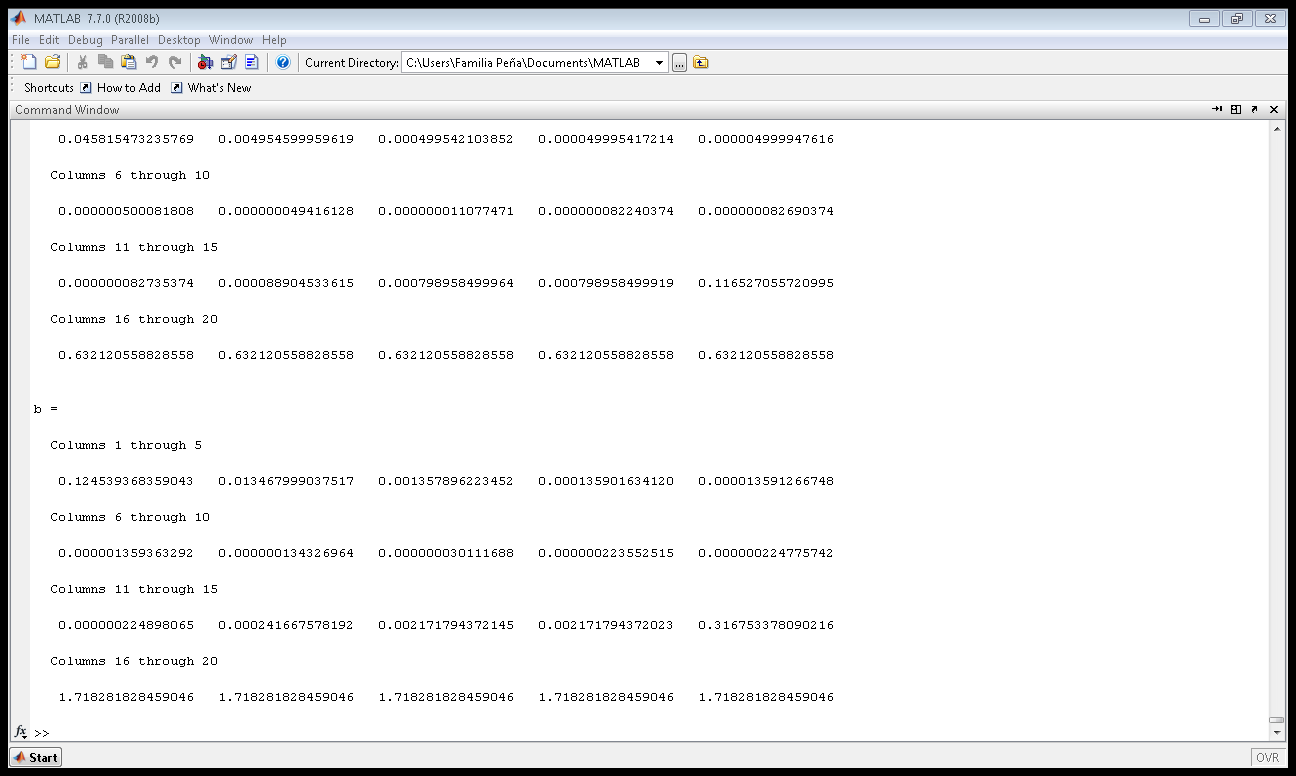
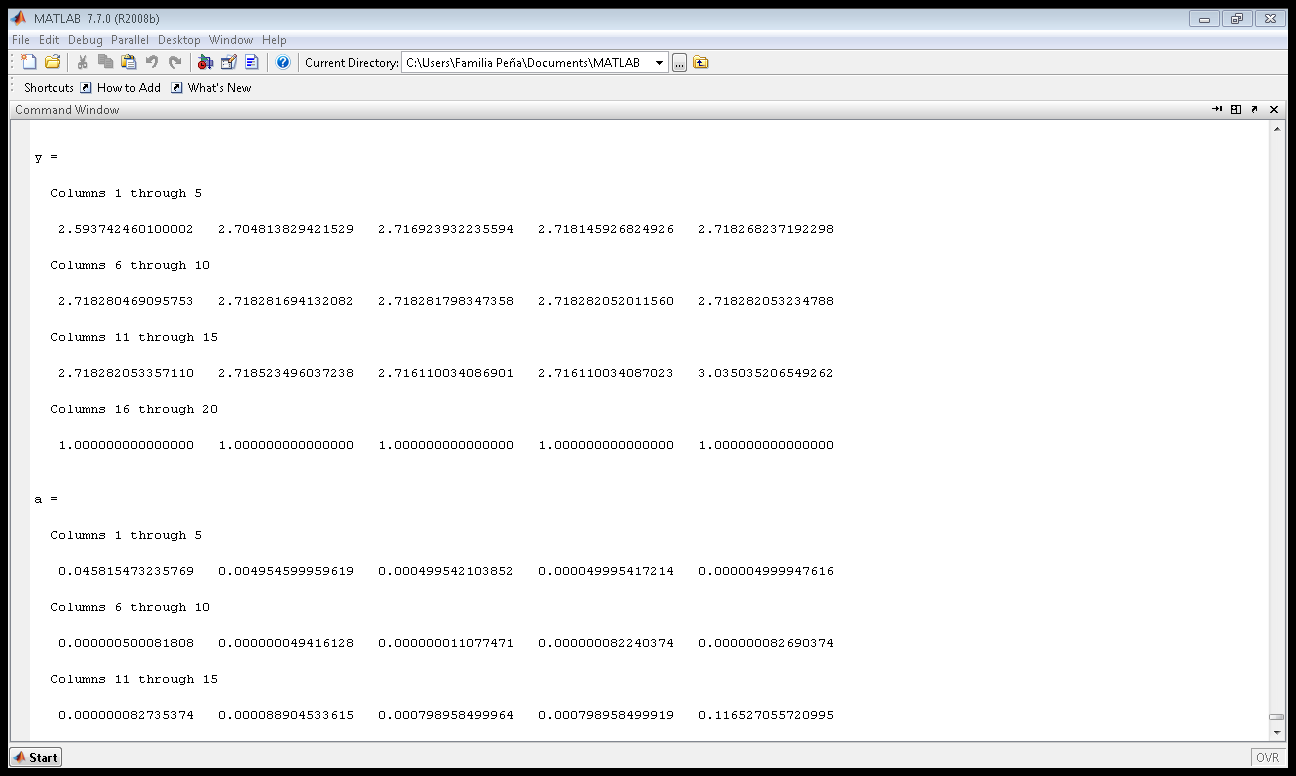
Específicamente, calcular (1 + 1/n) , para n= con k = 1,2, ...,20. Determinar el error en las aproximaciones sucesivas comparándolas con el valor de exp(1). El error siempre decrece cuando n crece? Explicar los resultados.



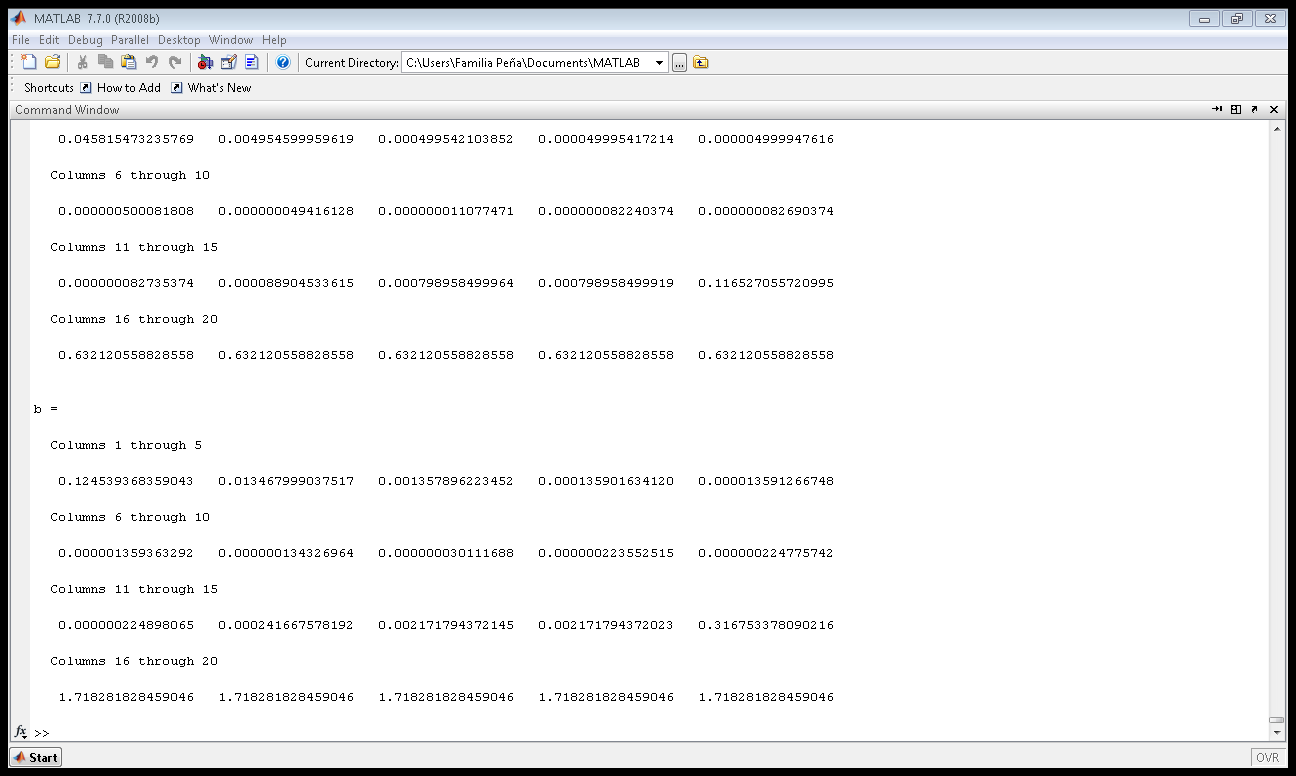
Aproximación por medio del límite a e:



Error Relativo:



Error Absoluto:



Observamos que el error aumenta conforme n crece, pues a partir de la iteración 16 el resultado del límite y por tanto de la aproximación a e, es 1, esto es debido a que se hace muy pequeño y la computadora al utilizar aritmética del punto flotante, ocurre en errores de redondeo y ya al ser un número cercano al lo redondea a cero, entonces 1+0 = 1. Creando diferencias entre el valor aproximado y el valor real de e, a esto lo conocemos como pérdida total.

3. Supongamos que queremos generar n+1 puntos equiespaciados en el intervalo [a,b], con espaciamiento h= (b-a)/n.

a. En aritmética de punto flotante cuál de los siguientes métodos es mejor y por qué?

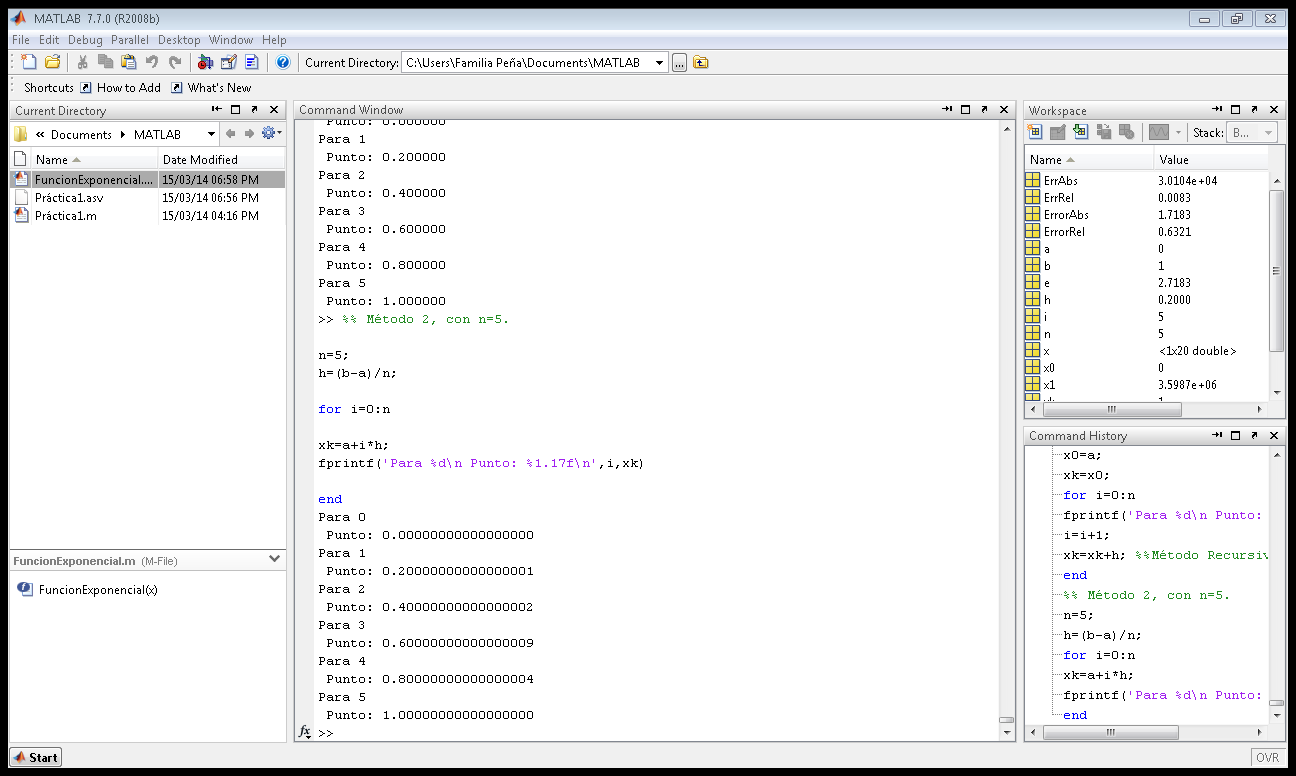
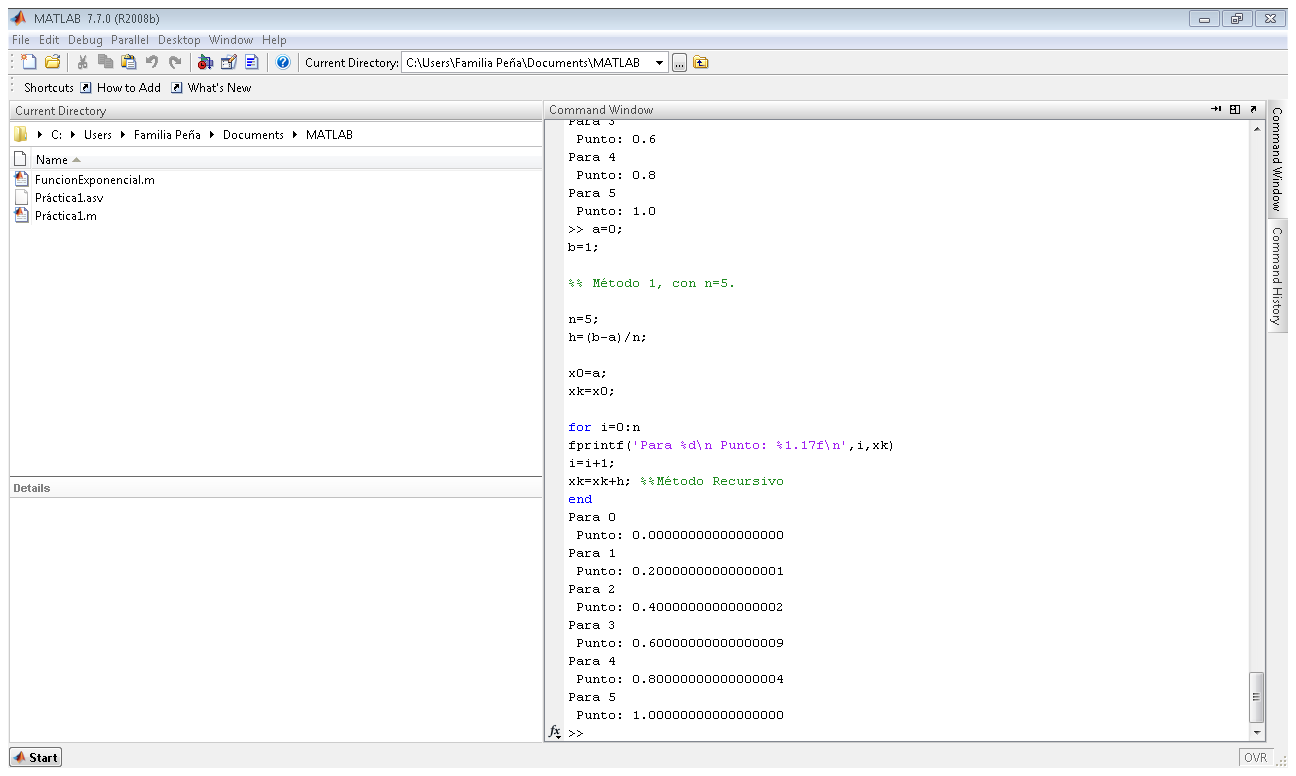
=a k=1,2,….n (1)

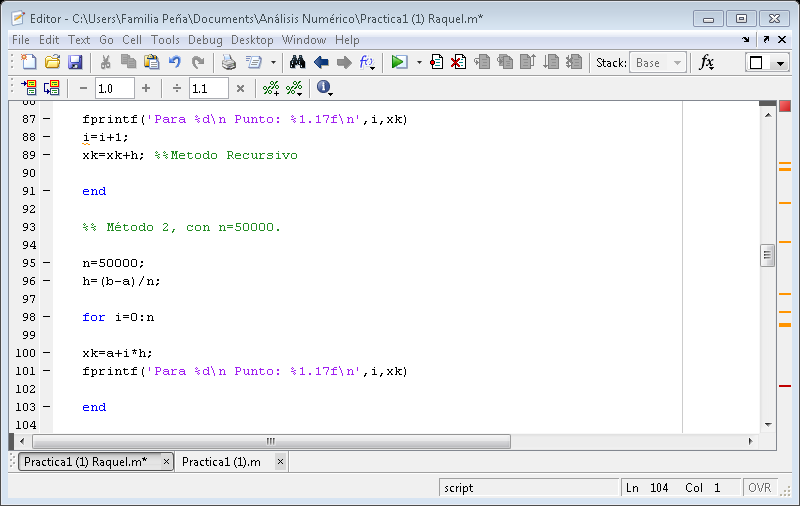
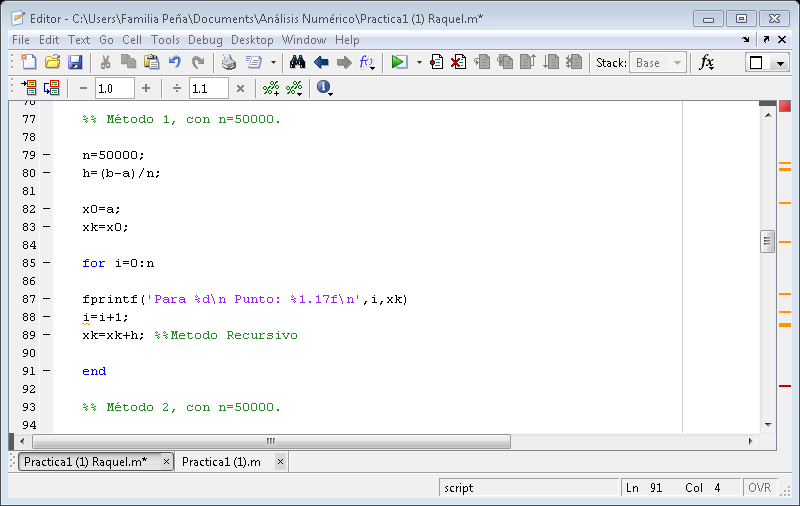
k=0,1,2,….n (2)

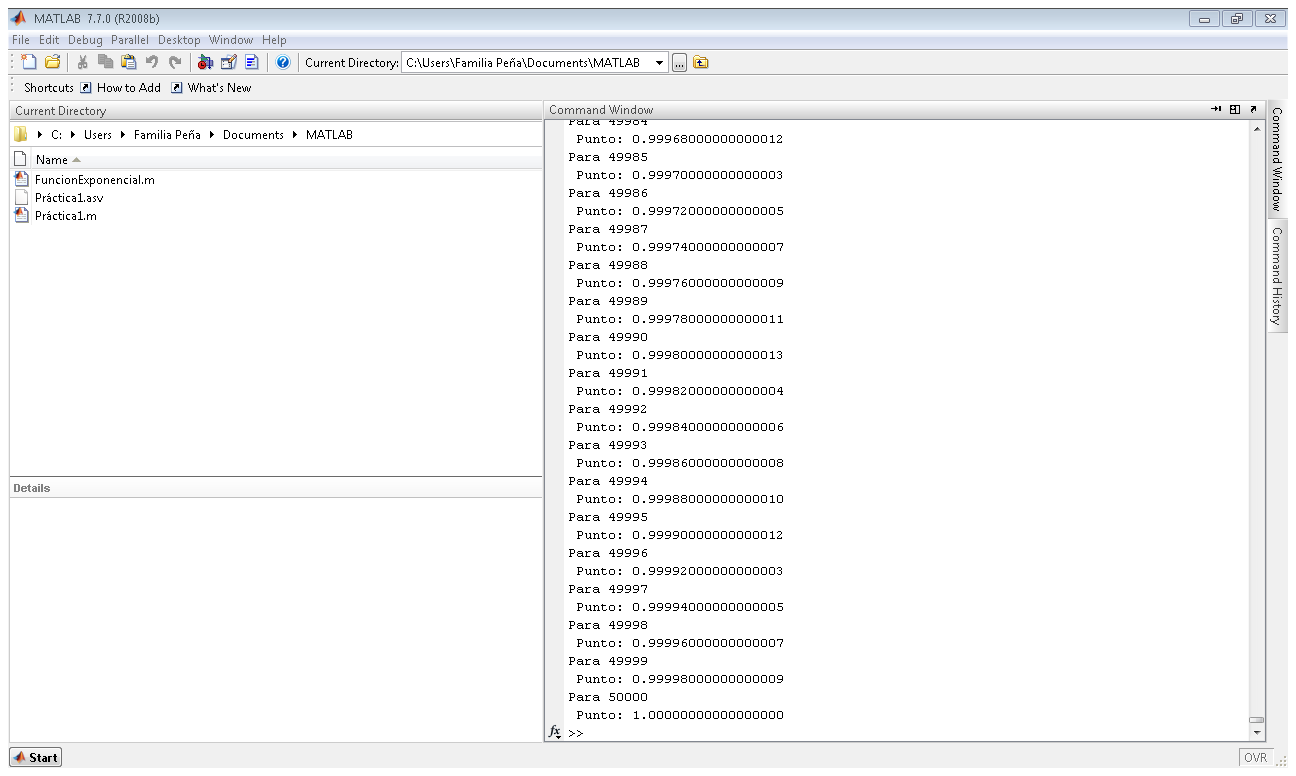
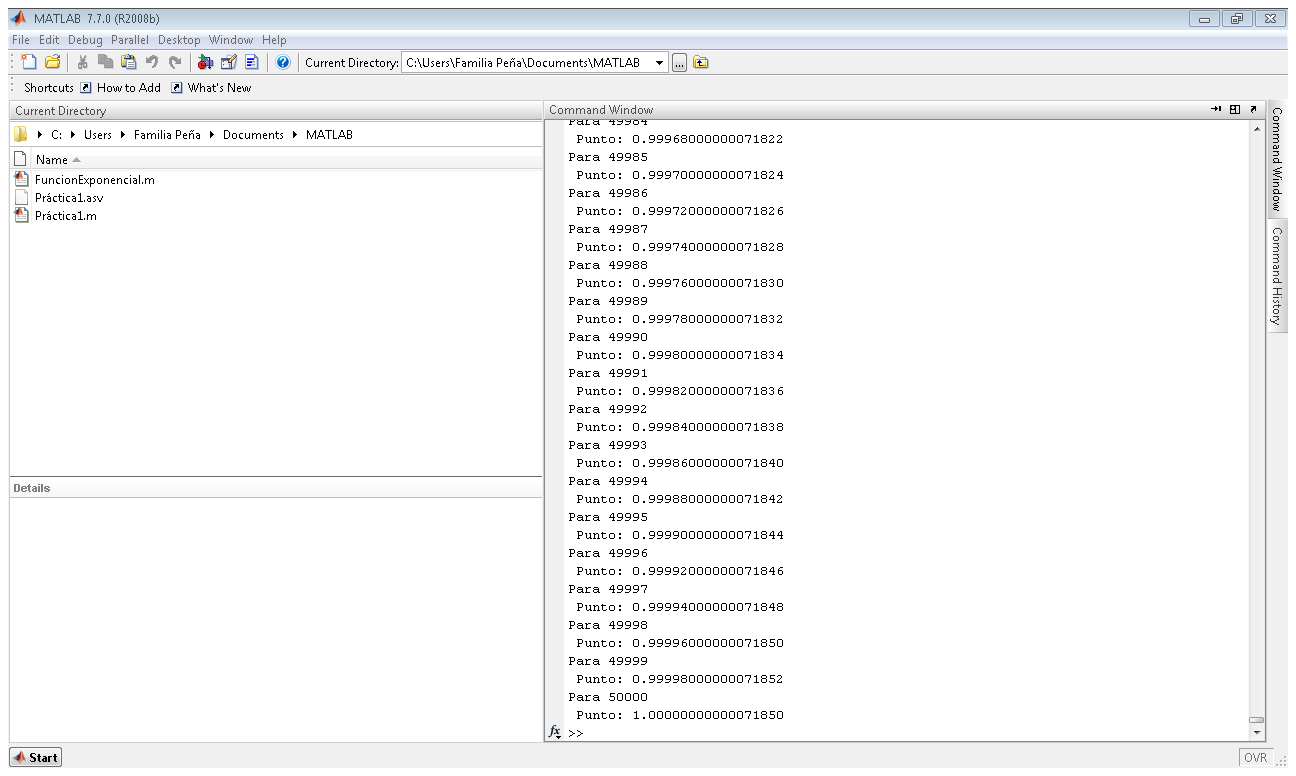
En aritmética del punto flotante el que más nos conviene es el (2). En primera instancia nos fijamos en el número de operaciones, que en este caso, el (1) sólo tiene suma, a diferencia del (2) que tiene suma y multiplicación, sabemos que entre más operaciones hagamos, nuestro cálculo perderá precisión.

Sin embargo, es importante observar que el primero tiene una forma recursiva, lo que implica que si tenemos un error, éste será arrastrado en todos los n pasos y éste es un factor mayor para decidir que el método más conveniente es el método (2).

b. Escribir un programa implementando ambos métodos con a=0 y b=1 para ilustrar las diferencias con n=5 n=50000







En efecto, para n más grande, el (1) hace un redondeo del 1, el cual va acarreando a través de la forma recursiva.

4. Considera la serie finita

a. Probar que la serie es divergente. Agrupar los términos en conjuntos conteniendo términos a para k=1,2,…

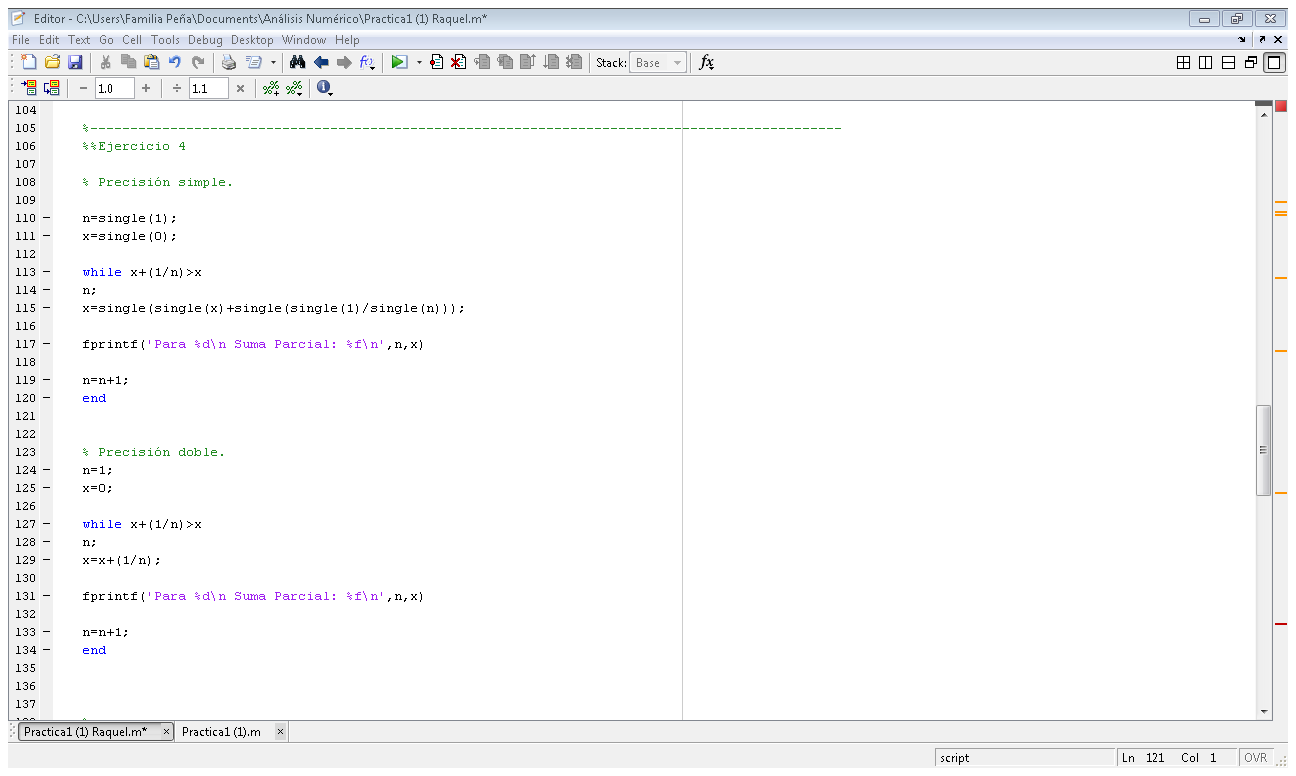
b. Explicar por qué sumando la serie en aritmética del punto flotante se obtiene una serie finita.

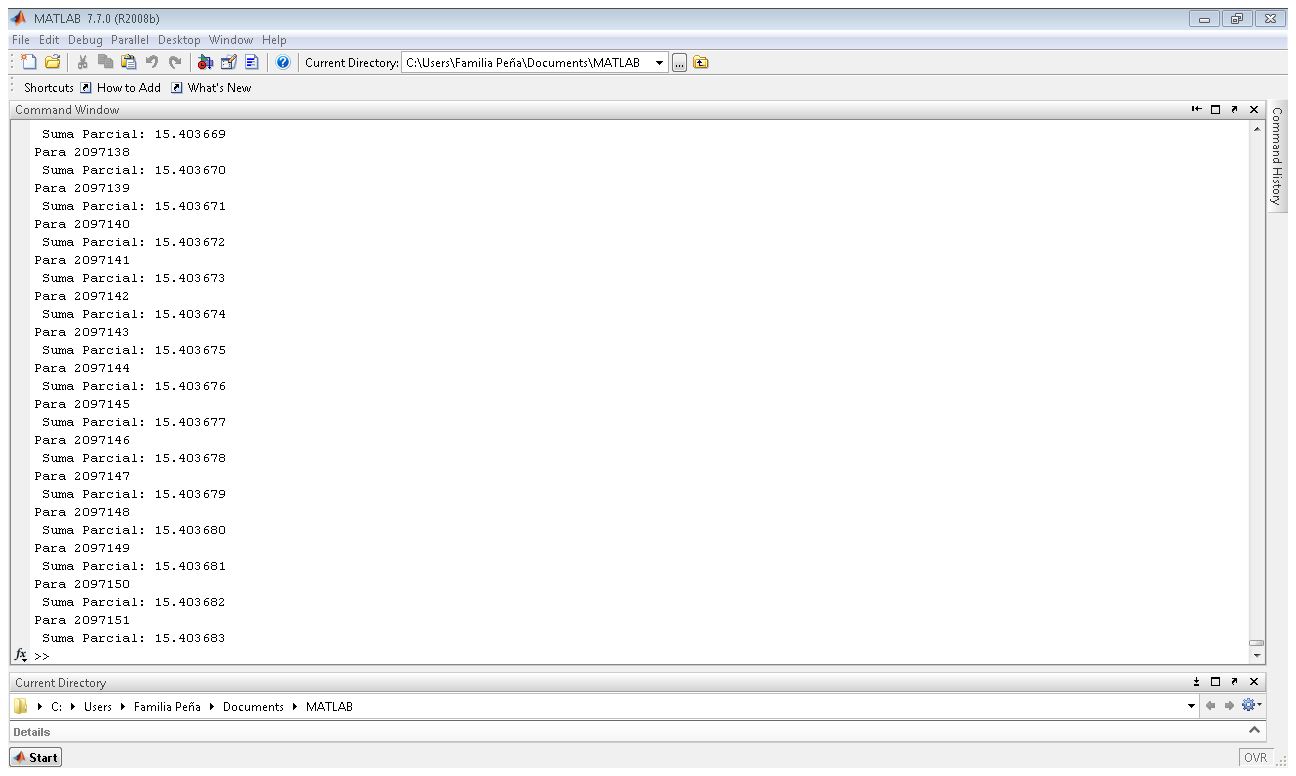
Observamos que si se tratara de aritmética exacta el ciclo sería infinito, pero como se trata de aritmética de punto flotante el ciclo llega al número más pequeño normalizado , es decir Nmin. Conforme n va creciendo, se va haciendo cada vez más pequeño hasta que < (en formato simple) cuando llega a este momento, el ciclo acaba ya que esté valor lo redondea a cero, convirtiéndolo en una serie finita pues deja de sumar elementos.

c. Tratar de predecir cuándo la suma parcial no cambiará en formato IEEE simple y doble.

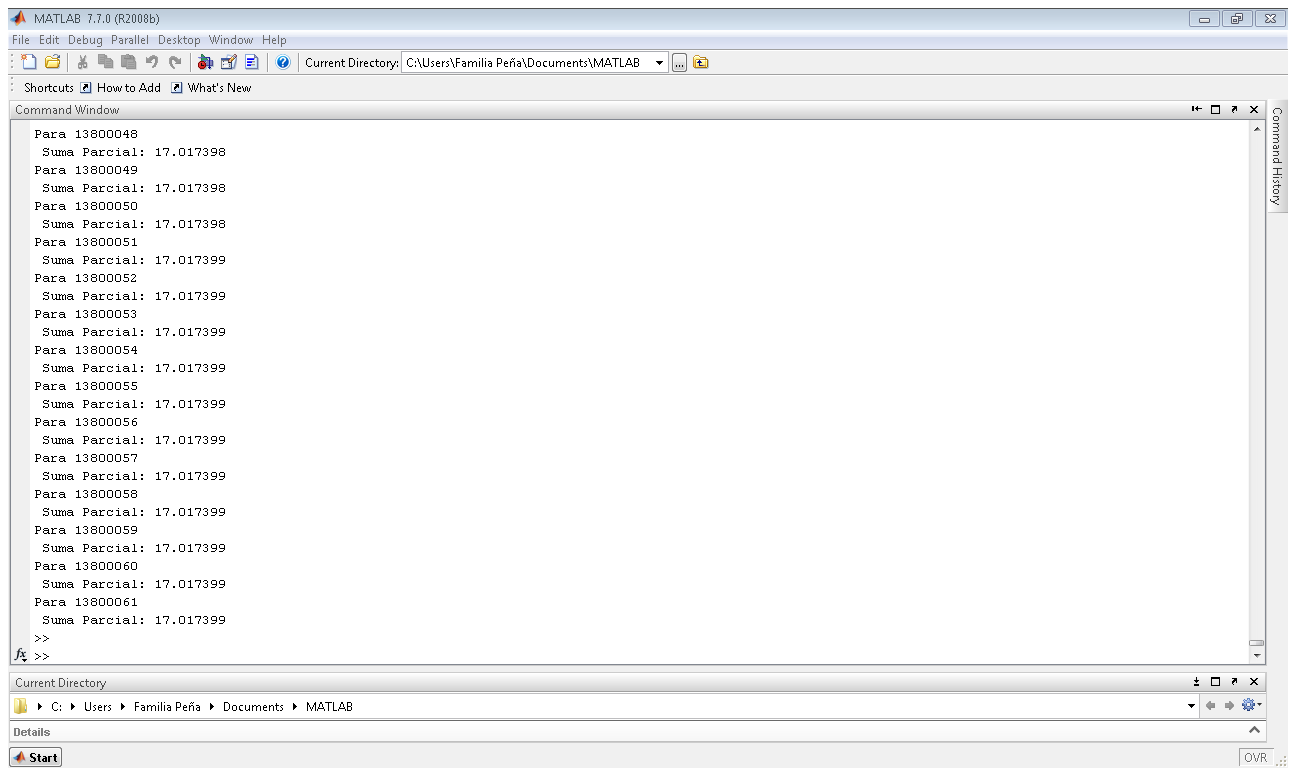
Como ya hemos notado anteriormente, se detendrá cuando se suficientemente despreciable, esto es, en formato de aritmética flotante, cuando <, entonces tenemos que n> . Esto es, para formato simple, la suma parcial ya no cambiará cuando n> y para formato doble, n>.

d. Escribir dos programas uno en precisión simple y el otro en precisión doble. Monitorear el progreso de la suma imprimiendo las sumas parciales y el índice n. ¿Cuál es el mejor criterio de paro que debería de usarse?





Para el formato simple observamos que Matlab se detiene en el índice 2 097,151. Indicando que la suma finita es: 15.403683.

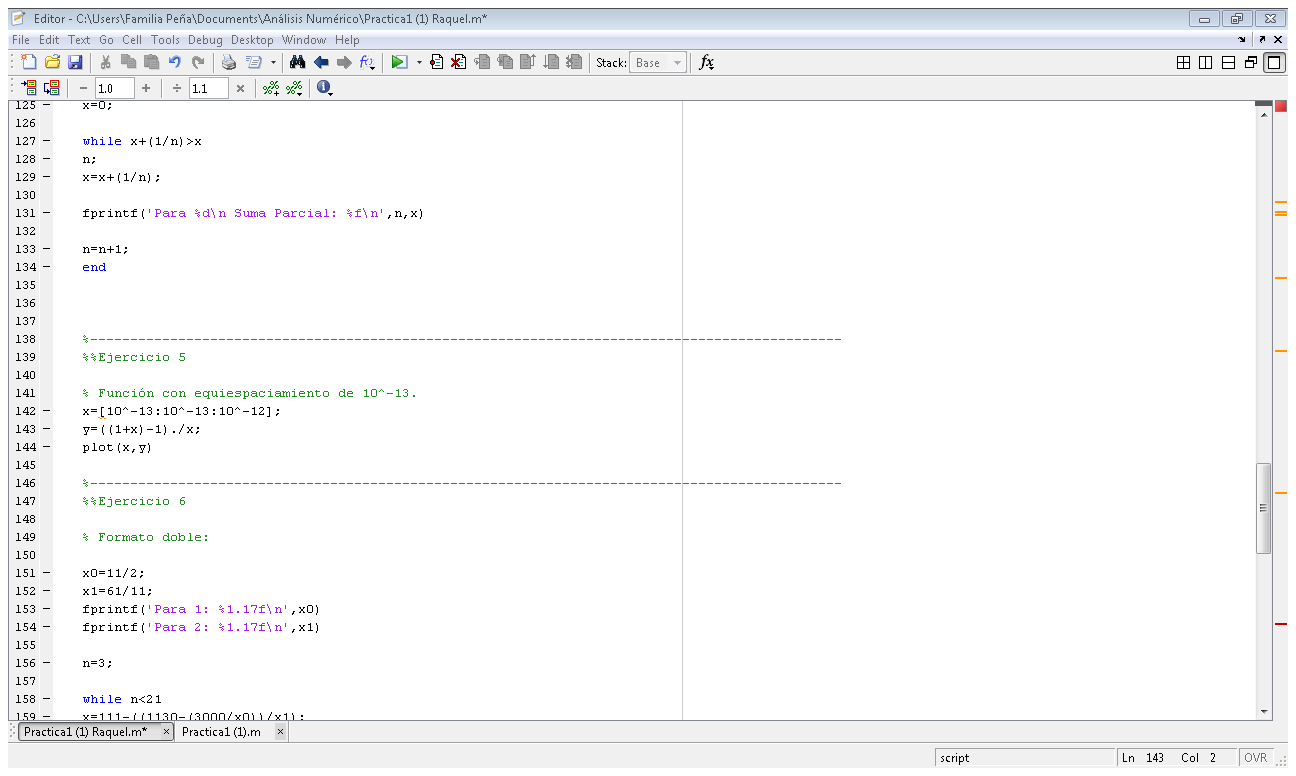


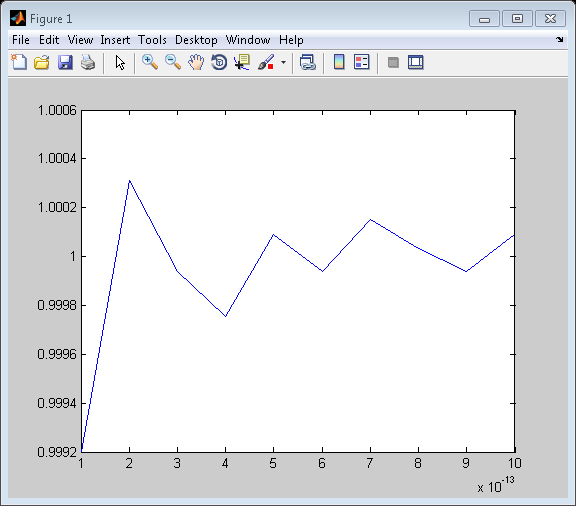
Notamos que la suma para el formato doble aún no termina, a pesar que la iteración ya es bastante grande. Como hemos visto en los anteriores incisos, en aritmética del punto flotante, ésta suma si es finita, por lo que creemos que también en formato simple se cumple. Lo único que bastaría checar el criterio de paro que se muestra a continuación.

El mejor criterio de paro es cuando el resultado de la suma de ya no difiera de su consecutivo, es decir cuando = , pues ya habíamos visto que lo redondeaba a cero.

5 a. Grafica la función f(x)= en [, usando cálculo.

b. Grafica la función anterior usando el programa con un esparcimiento de explicar los resultados.





En primera instancia, notamos que ésta gráfica no es nada parecida a la obtenida mediante cálculo. Observamos que al realizar la gráfica, el programa realiza tres operaciones: suma, resta y división. En cada una conlleva a un error debido al redondeo y además al hecho de tener un intervalo tan pequeño hace que para cada x (espaciamiento), el programa nuevamente haga un redondeo, provocando errores y pérdida de cifras significativas.

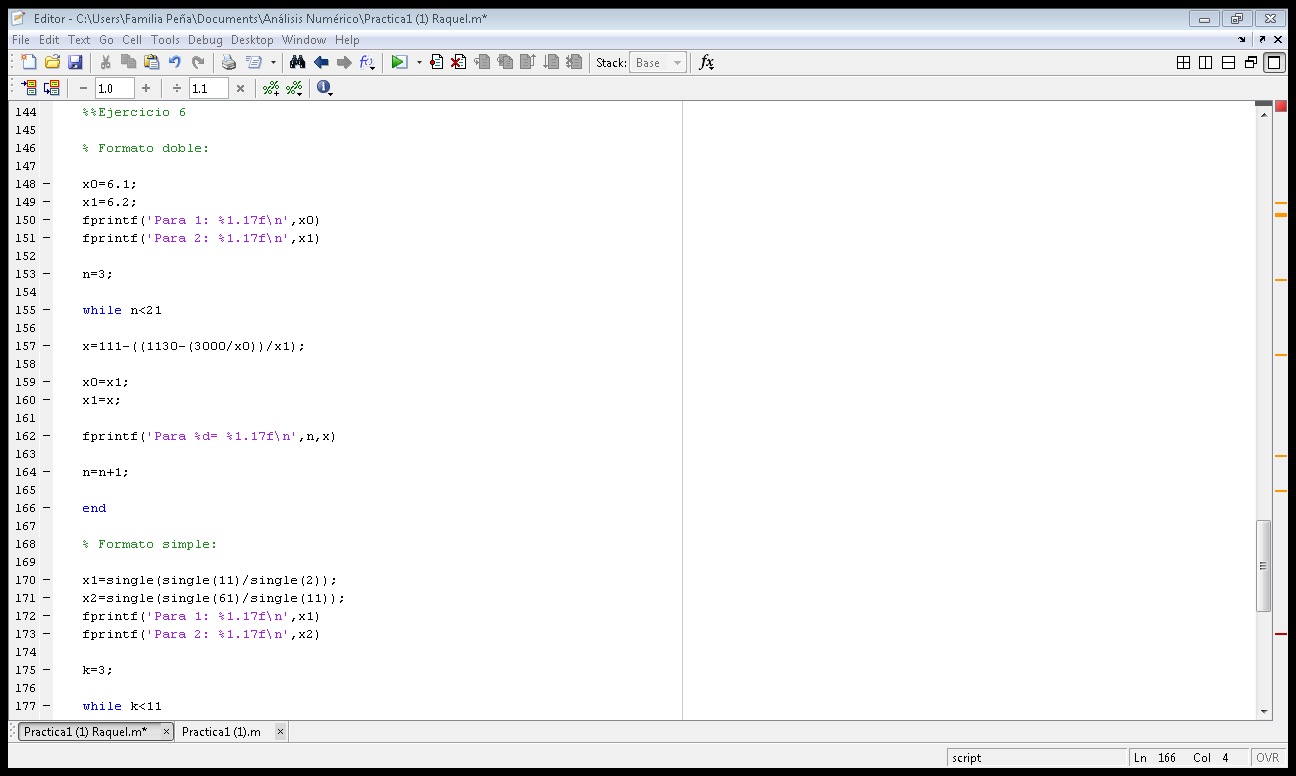
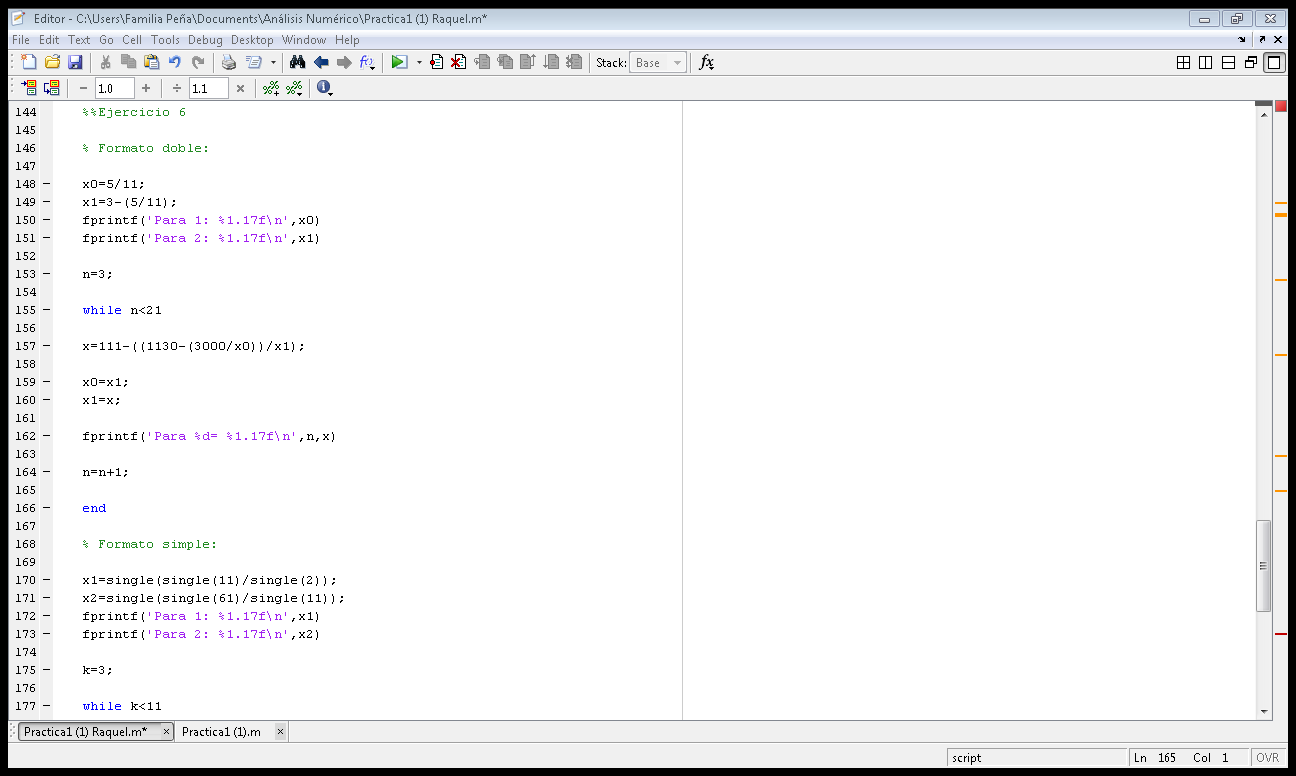
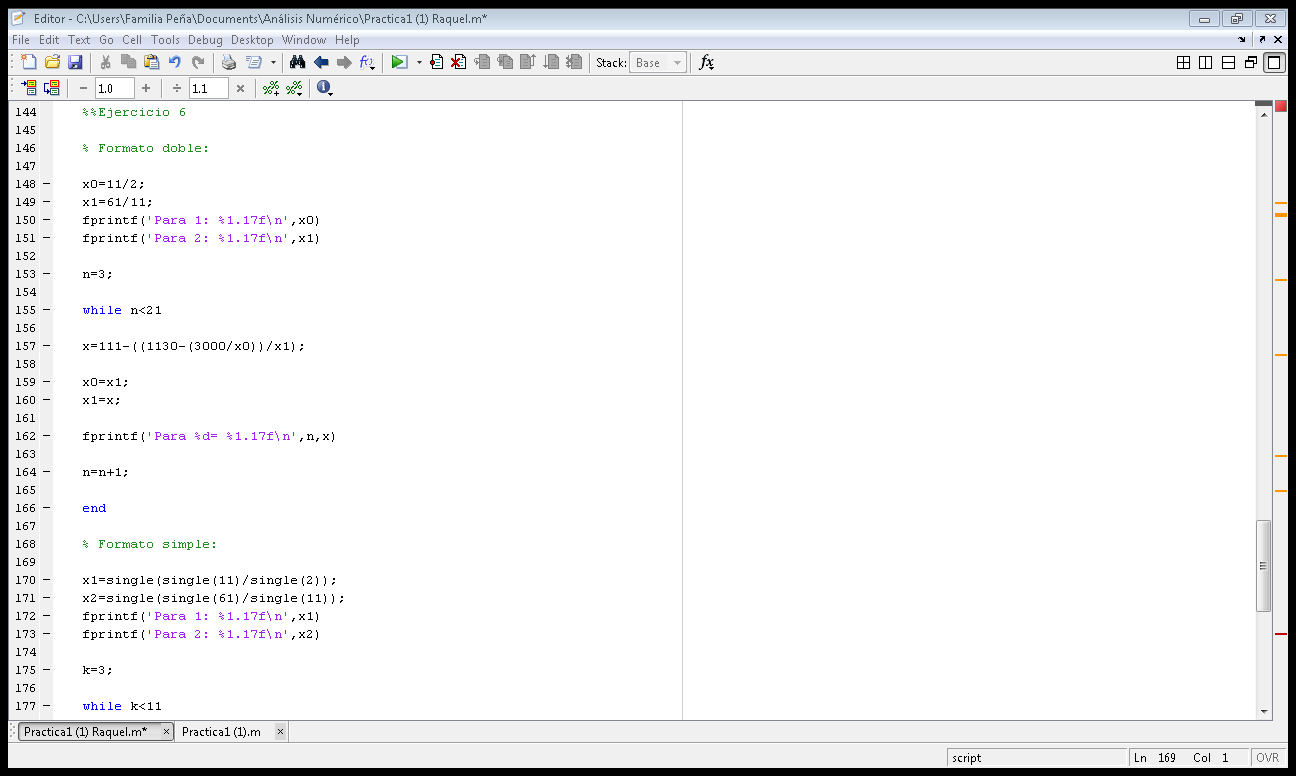
Entonces, en lugar de que la gráfica constante=1 se traslade, ocurre más de un error pues cada vez que se sustituye la x (una x pequeña) en la función, el programa hace un redondeo.

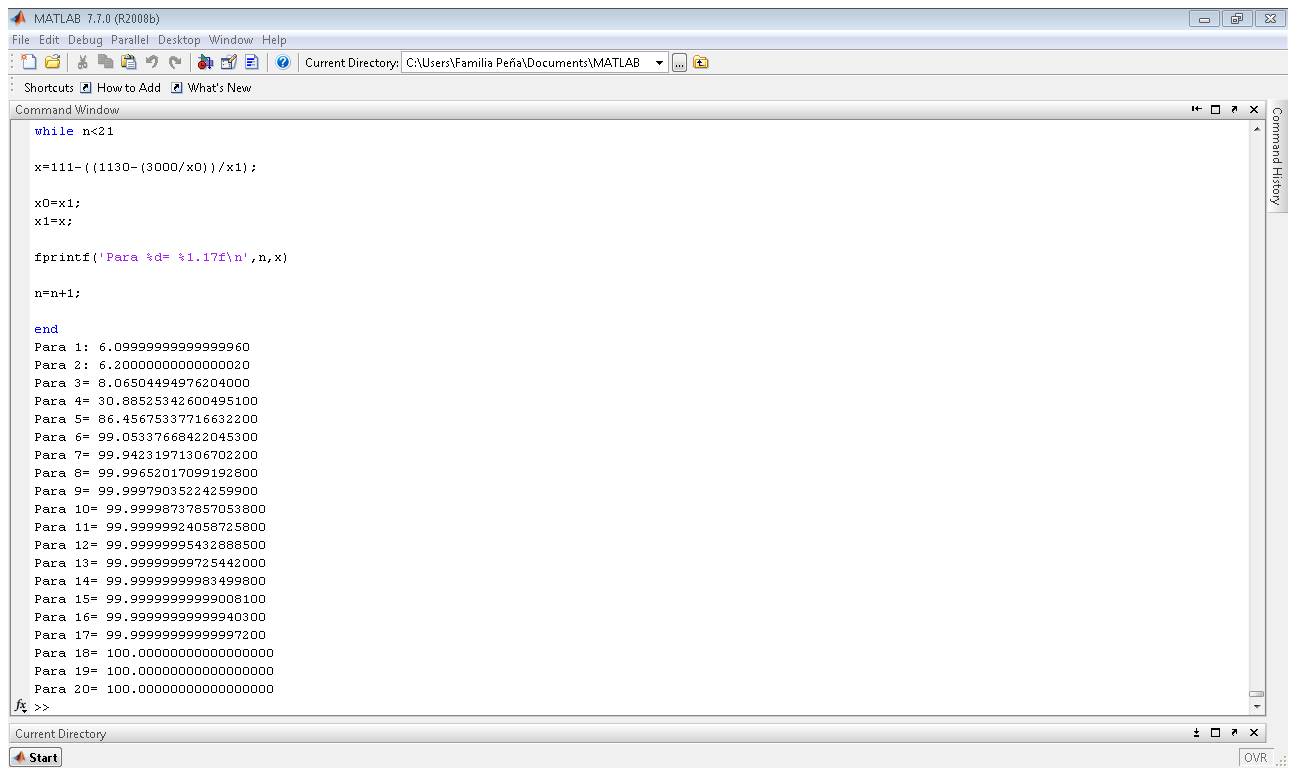
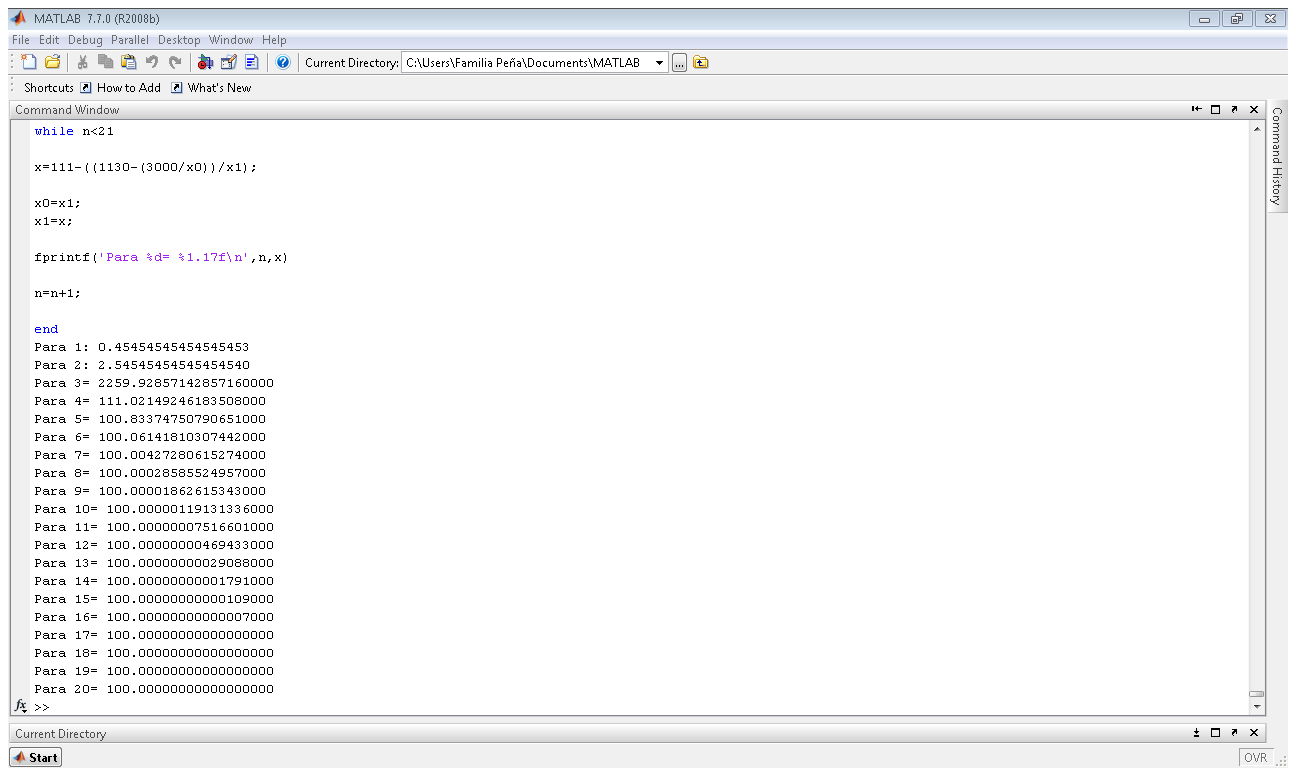
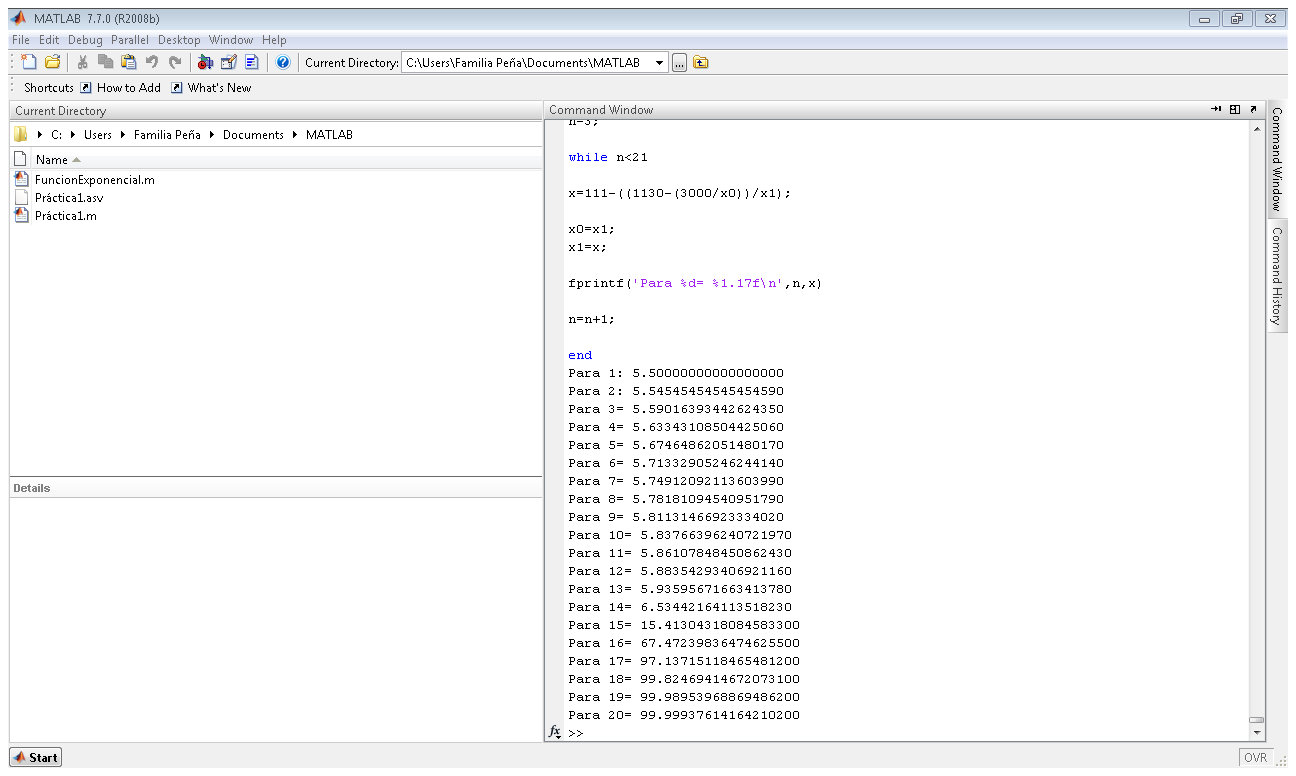
6. Escribir un programa que genere los primeros n términos en la sucesión dada por:

= Empezando con los valores = y =

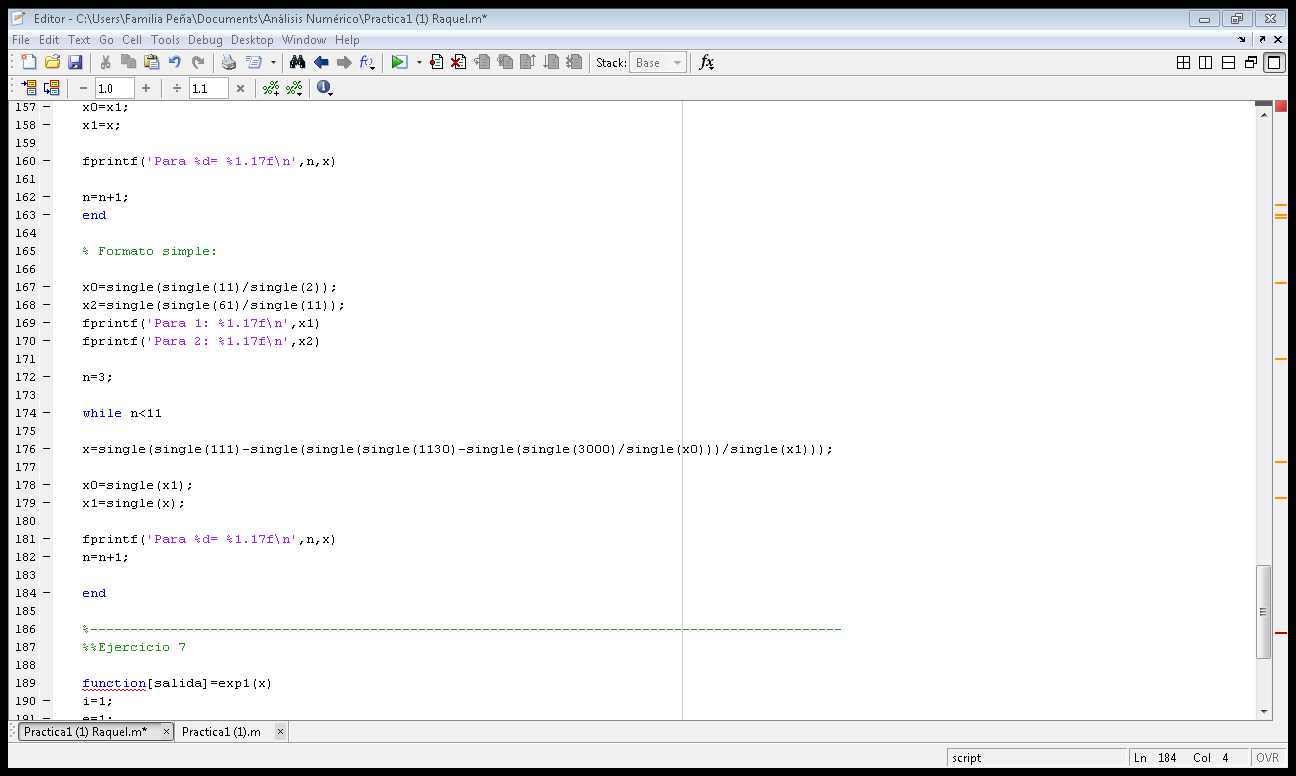
Usar n=10 para precisión sencilla y n=20 para precisión doble. La solución exacta es una sucesión monótona que converge a 6. Explicar los resultados.

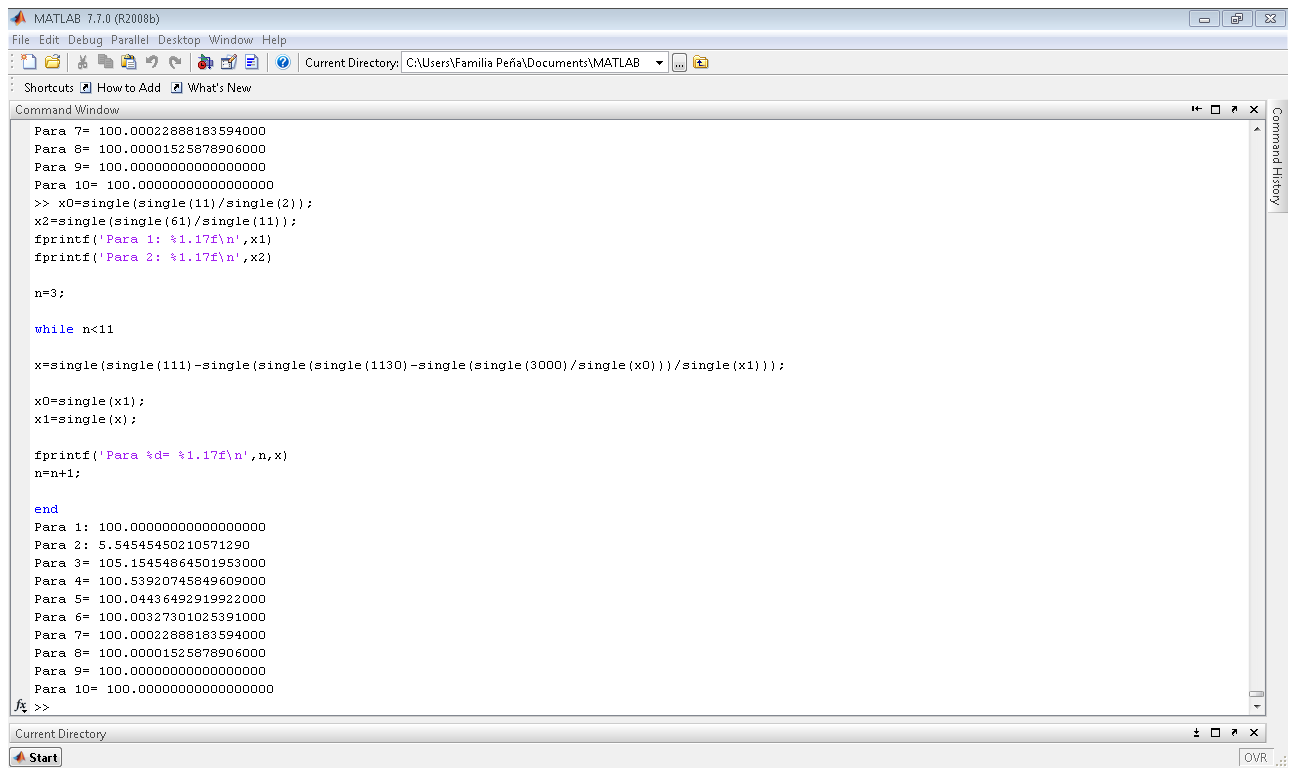
Formato doble:





Formato simple:





Observamos que al obtenemos 3 raíces del polinomio: x1=100, x2=5 y x3=6. Sin embargo, mediante el método de aritmética del punto flotante, la solución a la que converge es a 100, es decir, el 100 es numéricamente estable. Entonces las soluciones 5 y 6 no son numéricamente estables, pues no se convergen a ellas.

A pesar que cuando damos soluciones iniciales cercanas a 5 y cercanas a 6, para las primeras iteraciones si parece que convergen a éstos números, sin embargo, después cambian radicalmente para converger a 100. Con lo que reafirmamos nuestra postura acerca de la estabilidad de las soluciones.

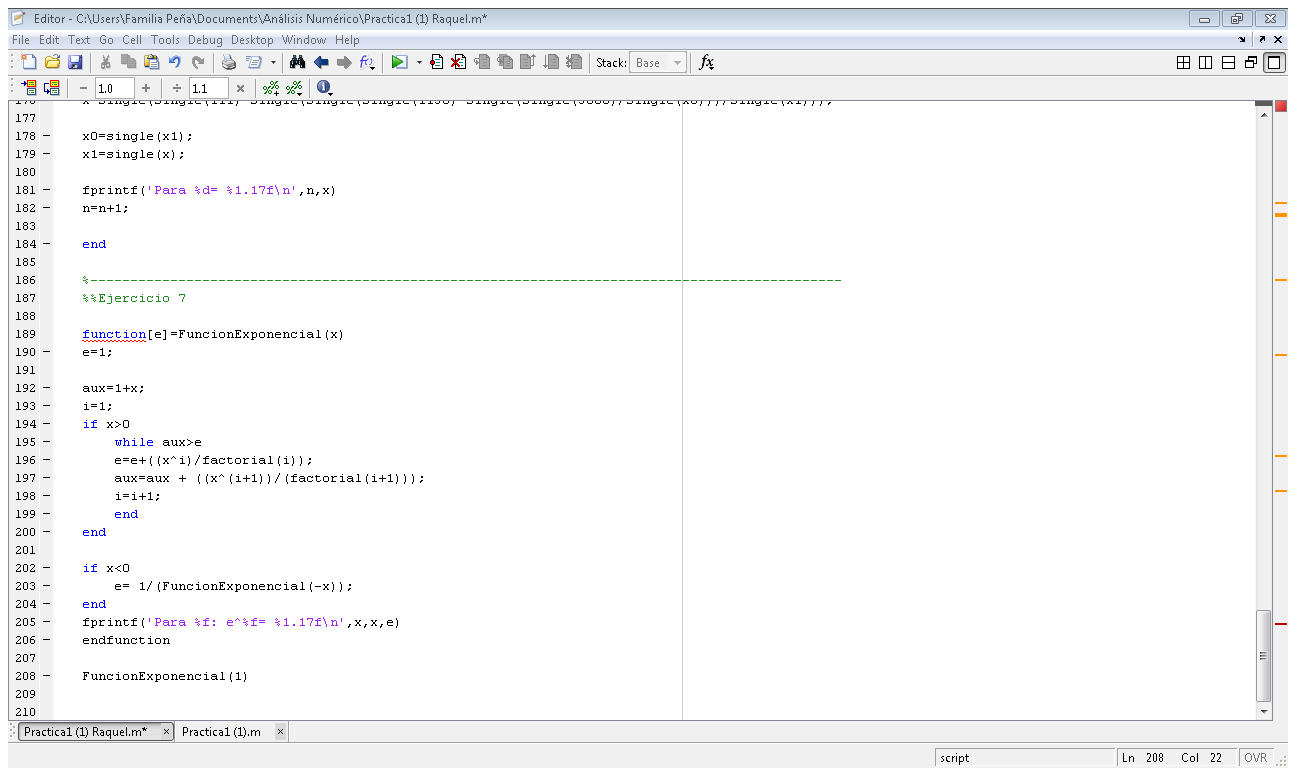
7. Escribir un programa que calcule la función exponencial

(a) Cual es el criterio que debe usarse para detenerse?

Es cuando encontremos una n tal que Sea despreciable en términos de aritmética flotante.

(b) Probar el programa para x = ±1, ±5, ±10, ±15, ±20, y comparar los resultados con la función implentada dentro de Matlab exp(x).

|  |  |  |
| --- | --- | --- |
| x | FuncionExponencial(-x) | exp(-x) |
| 1 | 2.7183 | 2.7183 |
| 5 | 148.41 | 148.41 |
| 10 | 2.2026E+06 | 2.2026E+06 |
| 15 | 3.2690E+06 | 3.2690E+06 |
| 20 | 4.85170E+08 | 4.85170E+08 |
| -1 | 0.36788 | 0.36788 |
| -5 | 0.0067379 | 0.0067379 |
| -10 | 0.0000454 | 4.54E-05 |
| -15 | 3.059E-07 | 3.059E-07 |
| -20 | 2.0612E-09 | 2.0612E-09 |



Notamos que con el programa, en el momento de evaluarlo en números negativos comienzan a existir errores, por ejemplo para x=-2 :

Entonces notamos que se van cancelando y empieza a arrastrarse un error debido a los signos alternantes. Por supuesto, difieren a los obtenidos mediante la función exp(x). Por lo tanto, para x negativos se procedió a hacer otra fórmula, explicada en el siguiente inciso.

(c) Se puede usar una serie en esta forma para obtener resultados precisos para x < 0?

Sí. Tenemos el siguiente desarrollo:

Entonces, si necesitamos obtener la función exponencial evaluada en un número negativo, en lugar de hacerlo como el anterior inciso que nos acarreaba errores, hay mayor precisión al calcular la función exponencial evaluada en el número positivo y obtener su inverso multiplicativo. Pues la función evaluada en dicho número positivo no nos traía errores.

Nota: Esta función se realizó en Octave pues hubo problemas al realizar funciones en Matlab. Todos los demás programas fueron implementados en Matlab.