Raquel Peña Alarcón

ANÁLISIS NUMÉRICO

Algebra lineal numérica e Interpolación

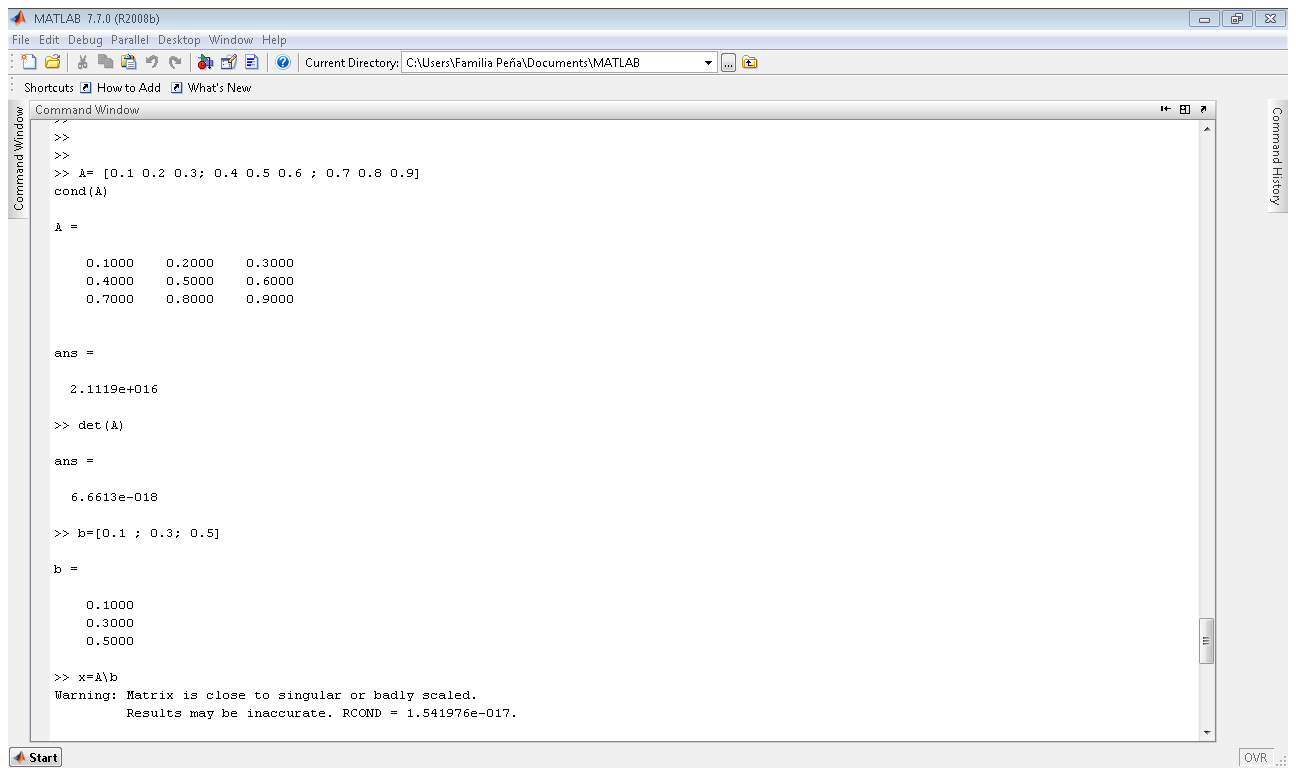
1. a) Mostrar que la matriz es singular. Describir el conjunto de soluciones al sistema Ax=n si b=A =

En aritmética exacta tenemos que :

b) Si usáramos eliminación Gaussiana con pivoteo parcial para resolver este sistema usando aritmética exacta, ¿En qué punto fallaría el proceso?

c) Como algunas de las entradas de A no son exactamente representables en un sistema de punto ﬂotante binario, la matriz ya no es singular cuando entra en una computadora. Entonces resolviendo el sistema por eliminación Gaussiana no necesariamente fallaría. Comparar la solución calculada con la descripción de la solución en el inciso (a). ¿Cuánto vale el número de condición=cond(A)?

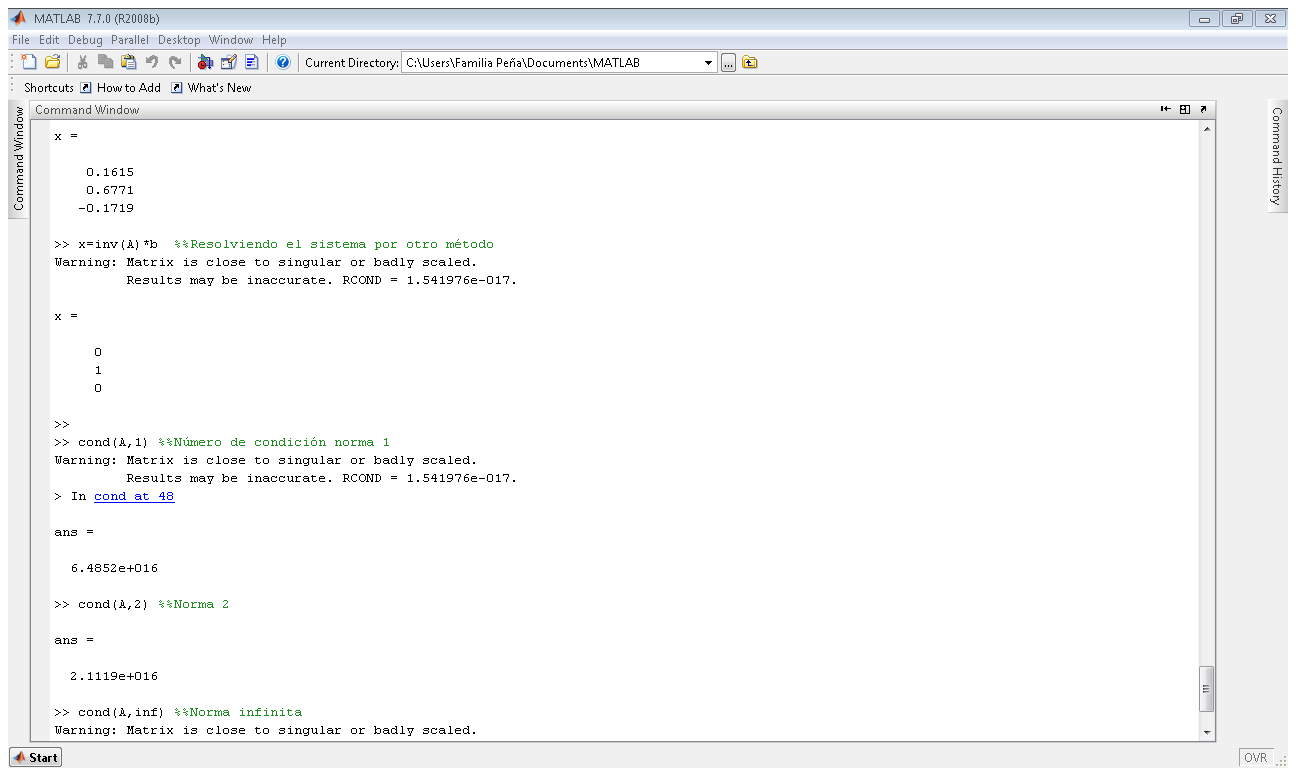
Notamos que en aritmética del punto flotante el determinante de A es distinto de cero, por lo que el sistema de ecuaciones tiene solución.

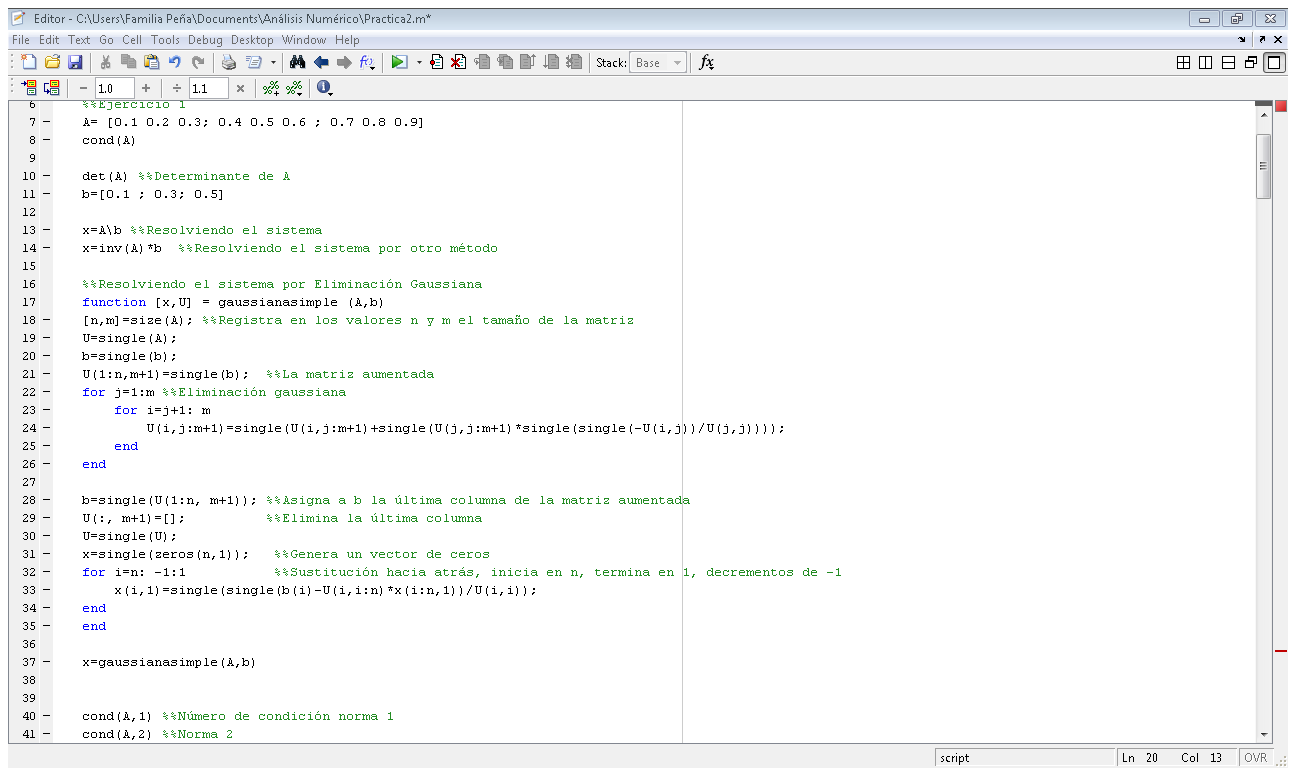


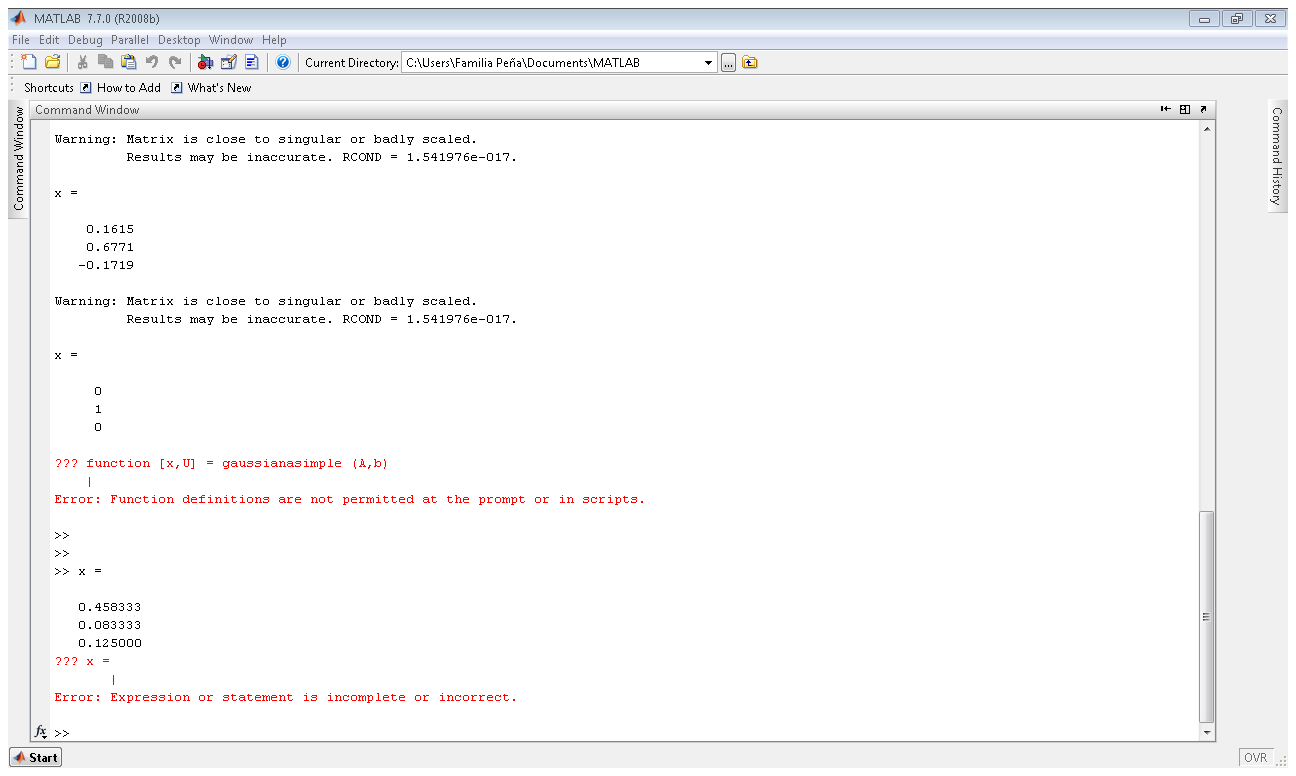
Observamos que el número de condición (norma 2) es muy grande, lo que nos podría indicar que la matriz A, está muy cercana a ser singular y además que es inestable. Cosa que notamos en aritmética exacta que sí es singular. El número de condición es un indicador sobre la precisión de las soluciones.

Notamos que el determinante en aritmética del punto flotante es muy pequeño, sin embargo, esto no nos indica si es A singular, sólo si es cero el determinante será singular.

Por medio del algoritmo x=A\b encontramos una solución al sistema de ecuaciones. Ahora, por medio del algoritmo de eliminación Gaussiana arroja otra solución para el sistema. Sin embargo, en aritmética del punto flotante teníamos que al ser el determinante distinto de cero, tenía una única solución, cosa que ha fallado.



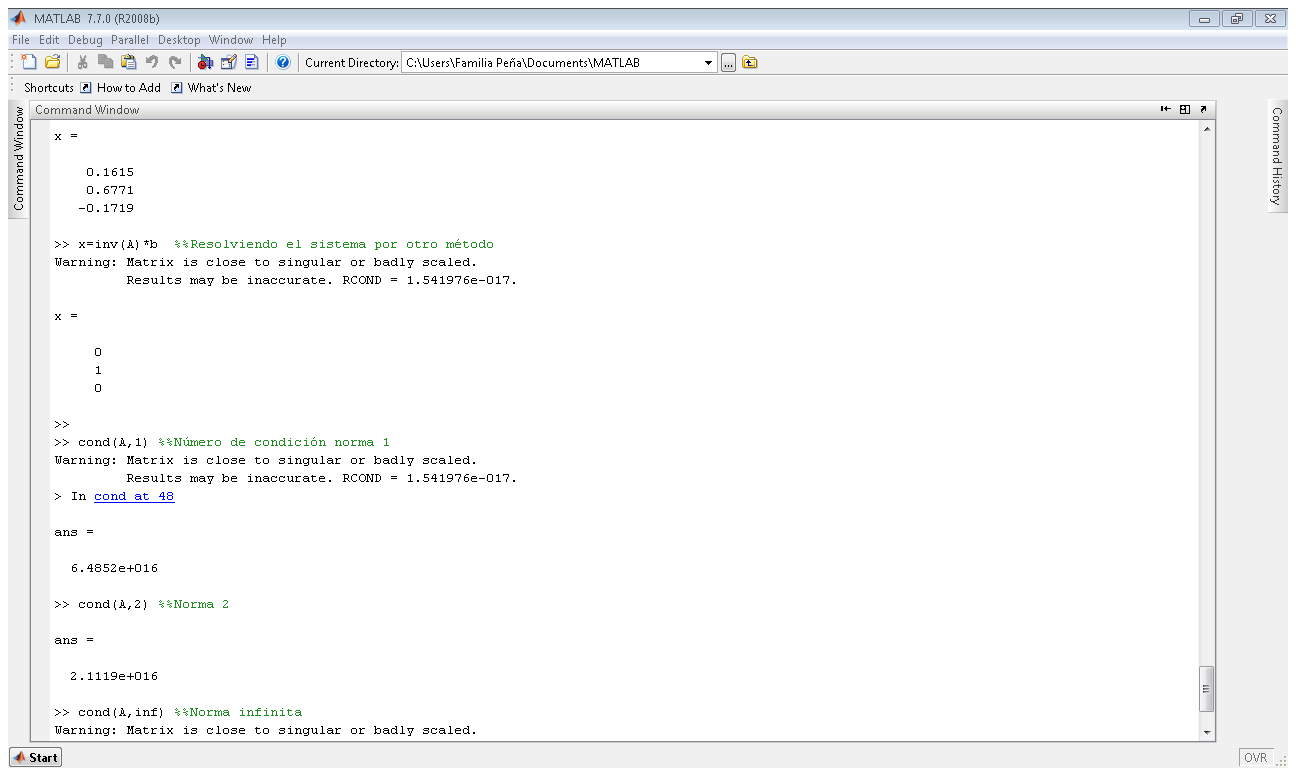




Entonces, comparándola con el inciso a) éstas dos soluciones satisfacen al Conjunto solución.

x=A\b

Eliminación Gaussiana

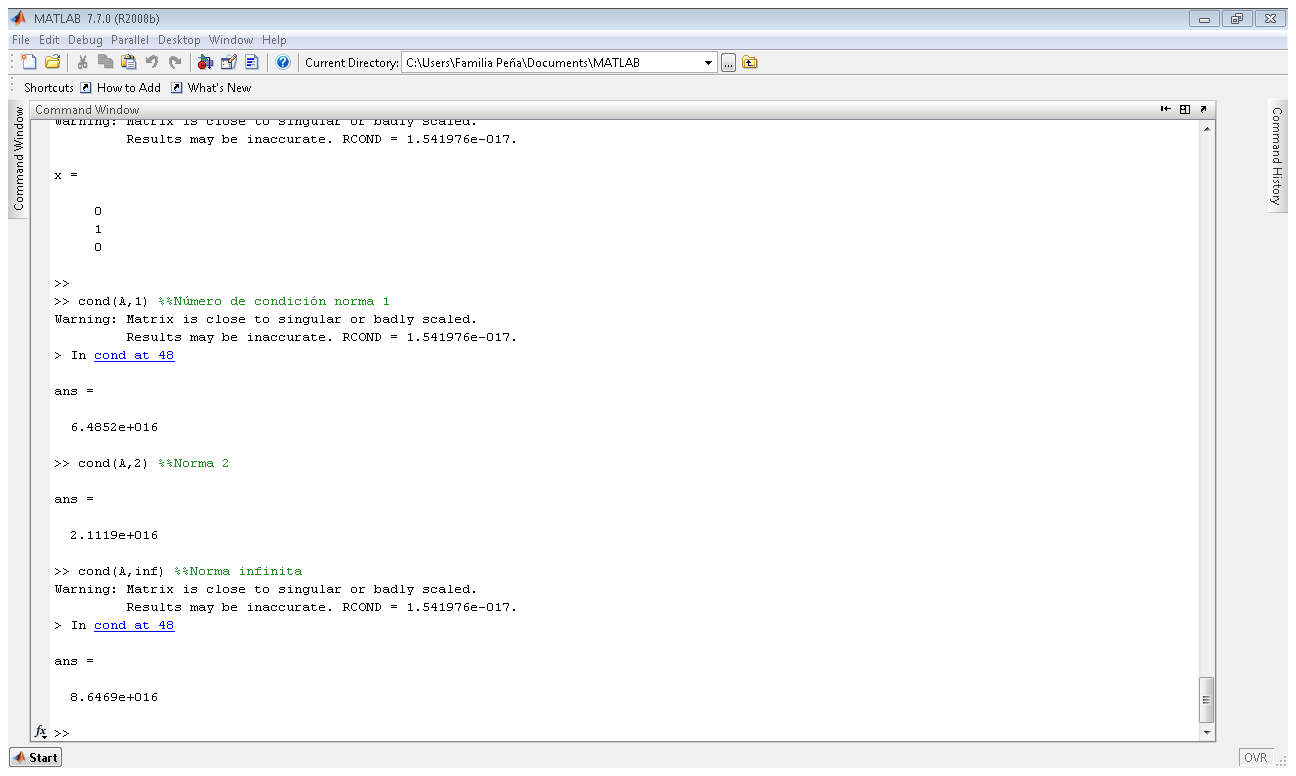


Por este método ,x=inv(A)\*b nos da una solución la cual no satisface el sistema de ecuaciones.

Posteriormente notamos las condiciones de acuerdo a su norma, que no importando la norma, está mal condicionada toda la matriz A.

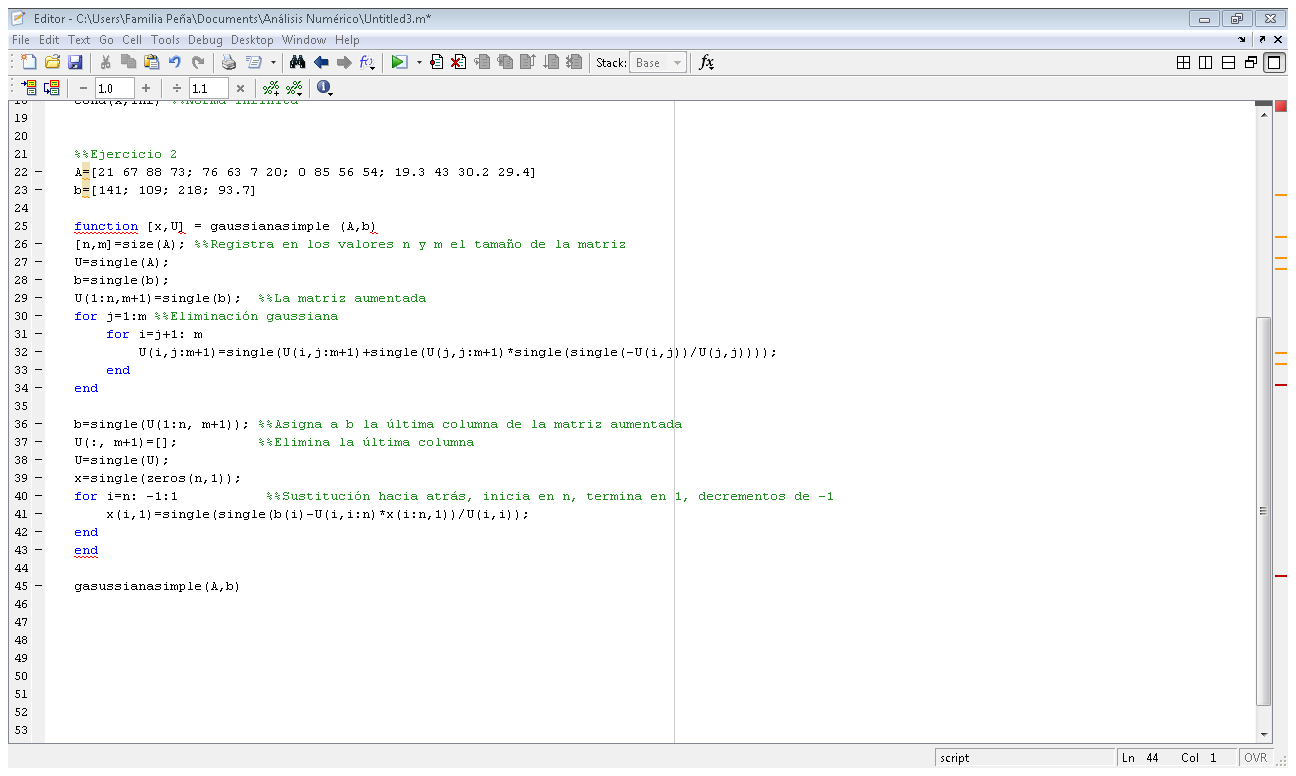
Además notamos que tenemos un problema mal condicionado el cual queríamos resolver con un algoritmo estable (eliminación Gaussiana), el resultado no es único.

El hecho que existan varias soluciones a pesar que habíamos visto que sólo debería de existir una, se debe a que está mal condicionada, reflejada en el número de condición bastante grande.



2.a) Usar una rutina de precisión simple para eliminación Gaussiana para resolver el sistema de ecuaciones Ax=b donde

A= b=



ans =

-0.99986

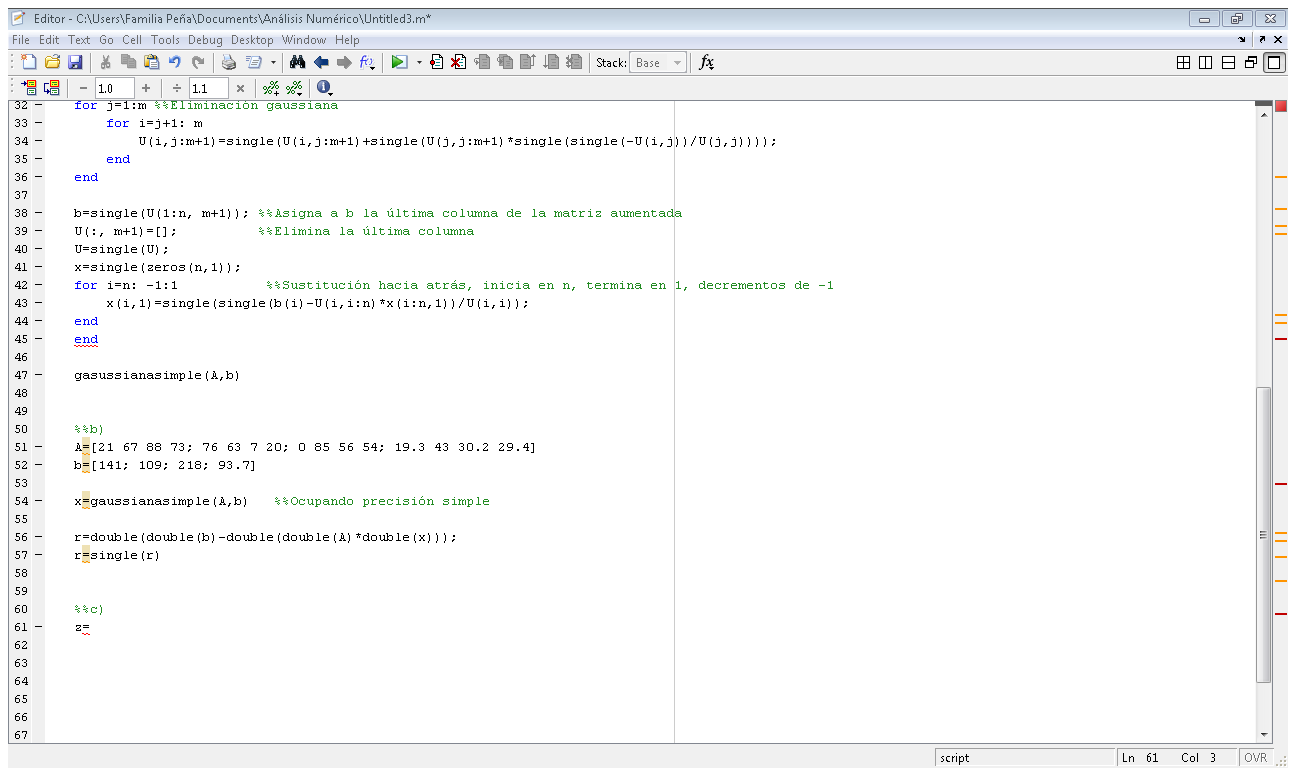
2.00065

-2.99773

3.99663

Tenemos que para precisión simple, la solución al sistema está dado por el vector anterior.

b) Calcular el residual r = b − Ax usando artimética de precisión doble, pero almacenando el resultado r en precisión simple. Nótese que la rutina empleada para resolver el sistema puede destruir el arreglo que contiene a la matriz A, es posible que se tenga que guardar una copia por separado para poder calcular el residuo.



r =

2.1875e-005

2.4319e-005

-1.2398e-005

2.0999e-005

Observamos que en un principio calculamos todo con precisión doble y después se define con precisión simple. En primera instancia no notamos que sea tan notorio el cambio en r, sin embargo, posteriormente se en las operaciones se verá reflejado dicho cambio.

c) Resolver el sistema de ecuaciones Az= r para obtener una solución mejorada x+z

Tenemos la solución sin mejorar:

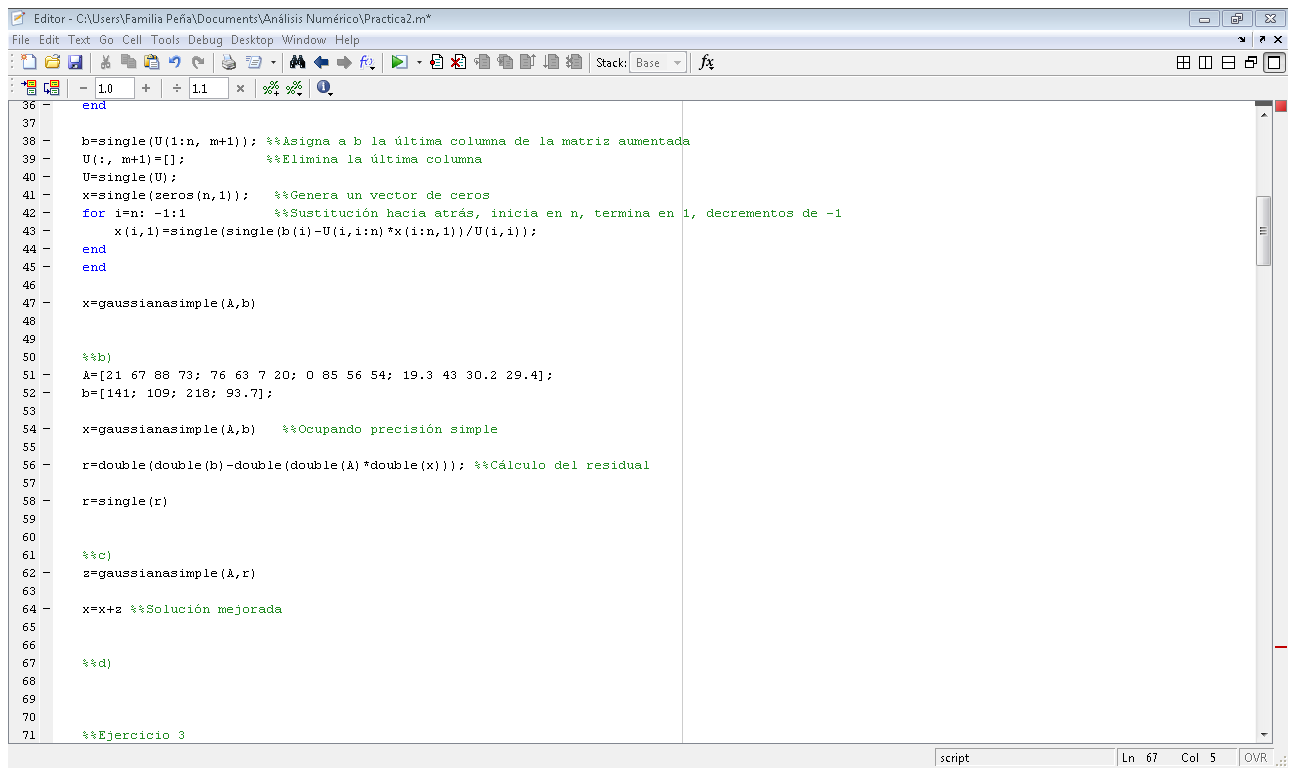
x=

-0.99986

2.00065

-2.99773

3.99663



r =

2.1875e-005

2.4319e-005

-1.2398e-005

2.0999e-005

z =

-1.4237e-004

-6.4648e-004

-2.2725e-003

3.3740e-003

x =

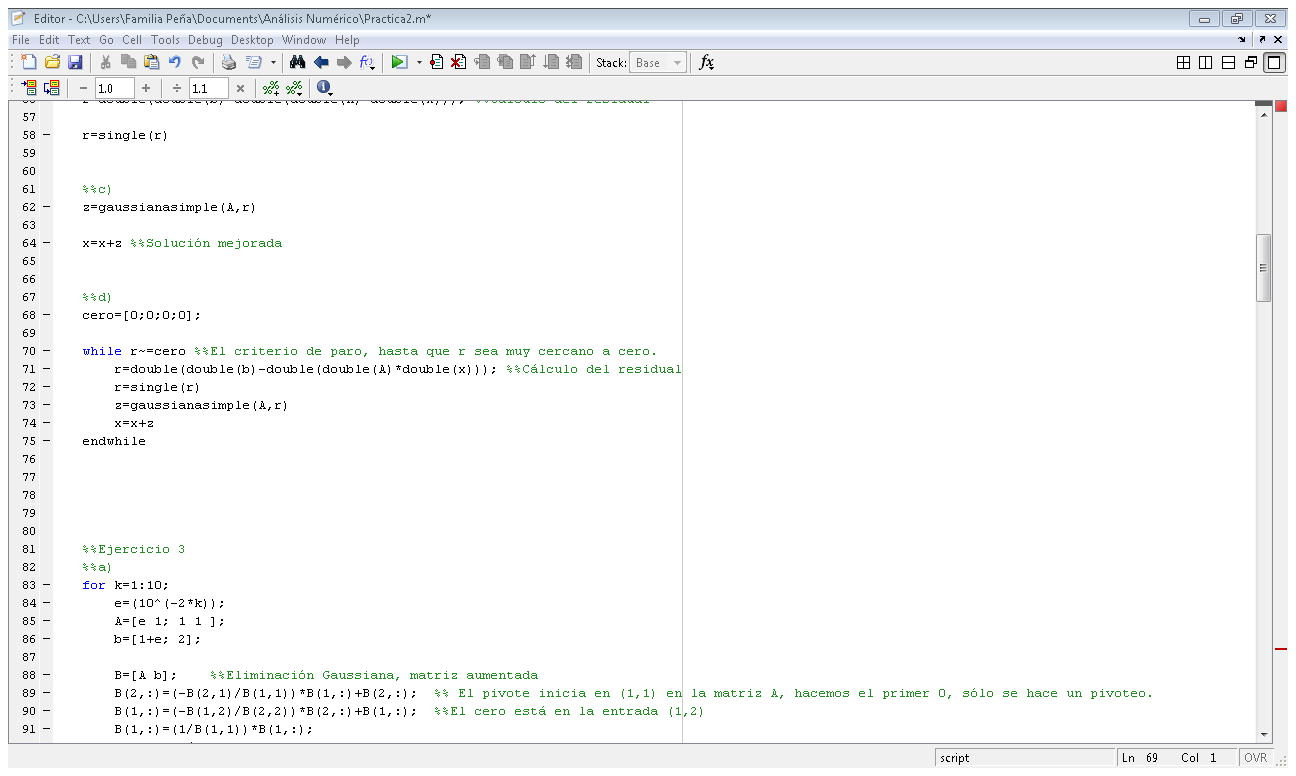
-1.0000

2.0000

-3.0000

4.0000

d) Repetir los pasos en b) y c) hasta que no se observe mejoría



r =

-7.1526e-006

6.1989e-006

Para la penúltima iteración, observamos que el residuo se hace cada vez más pequeño. El resultado x, se hace más exacto.

-9.5367e-007

-3.8147e-007

z =

2.2088e-012

1.0277e-011

-4.7680e-007

4.7678e-007

x =

-1

2

-3

4

r =

Para esta última iteración el residuo ya es cero. Mejorando aún más la solución ya que r=b-Ax, lo que se hace para mejorar es Az=r entonces Az=b-Ax Az+Ax=b  A(z+x)=b  AX=b donde X=z+x en donde encontramos una solución X mejorada.

Entonces parece razonable que r y z sea cero. Y tenemos la solución mejorada X. sin embargo notamos que esta solución no cambió mucho a la obtenida anteriormente, sólo r y z cambiaron.

0

0

0

0

z =

0

0

0

0

x =

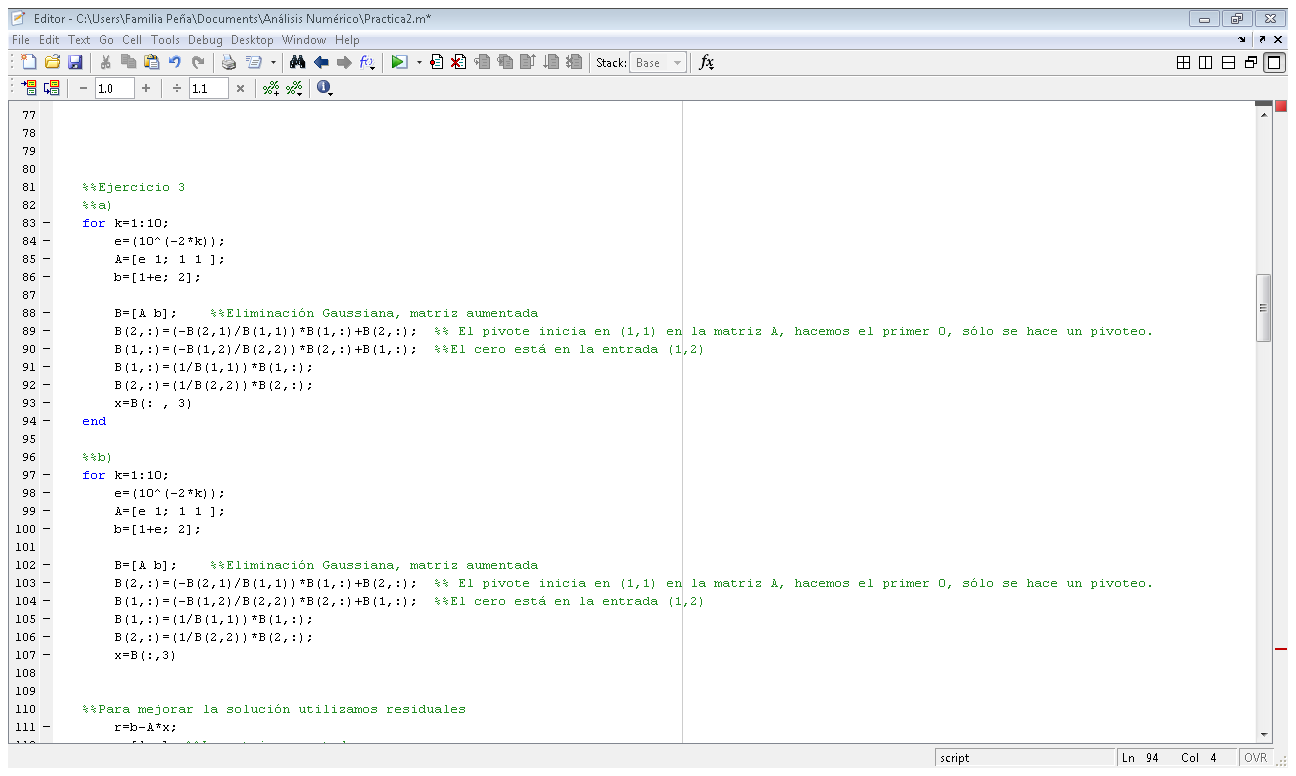
-1

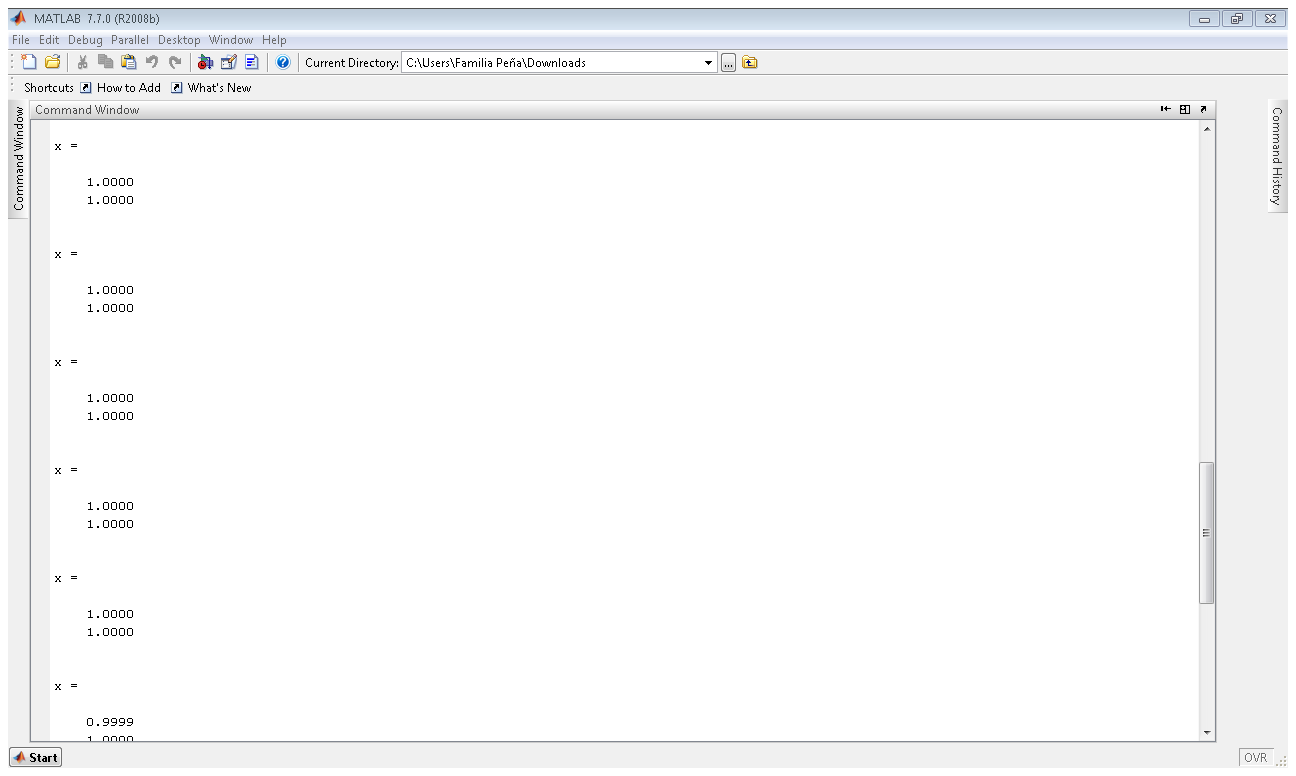
2

-3

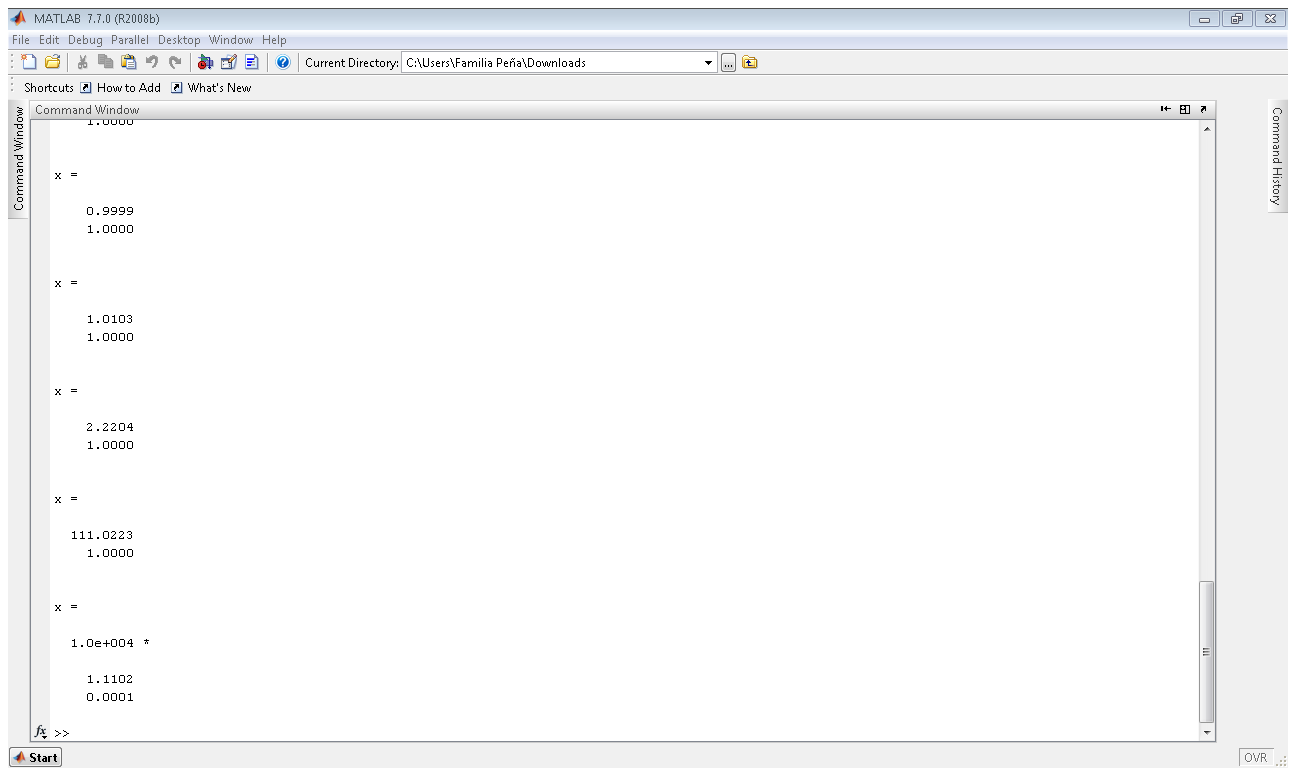
4

3. a) Usar eliminación Gaussiana sin pivoteo para resolver el sistema lineal de ecuaciones para k=1,…,10. La solución exacta es x= independiente del valor de . ¿Cómo se comporta la precisión de la solución cuando decrece?





Notamos que a diferencia de lo planteado en este inciso, la solución x si cambia de acuerdo al valor que tome . Observamos que para los primeros valores de , la solución parece estable y es la misma a la sugerida en el planteamiento del problema, es decir, la exacta la cual es x= . Es decir, para cuando no es tan pequeño, la solución es la misma a la de aritmética exacta.

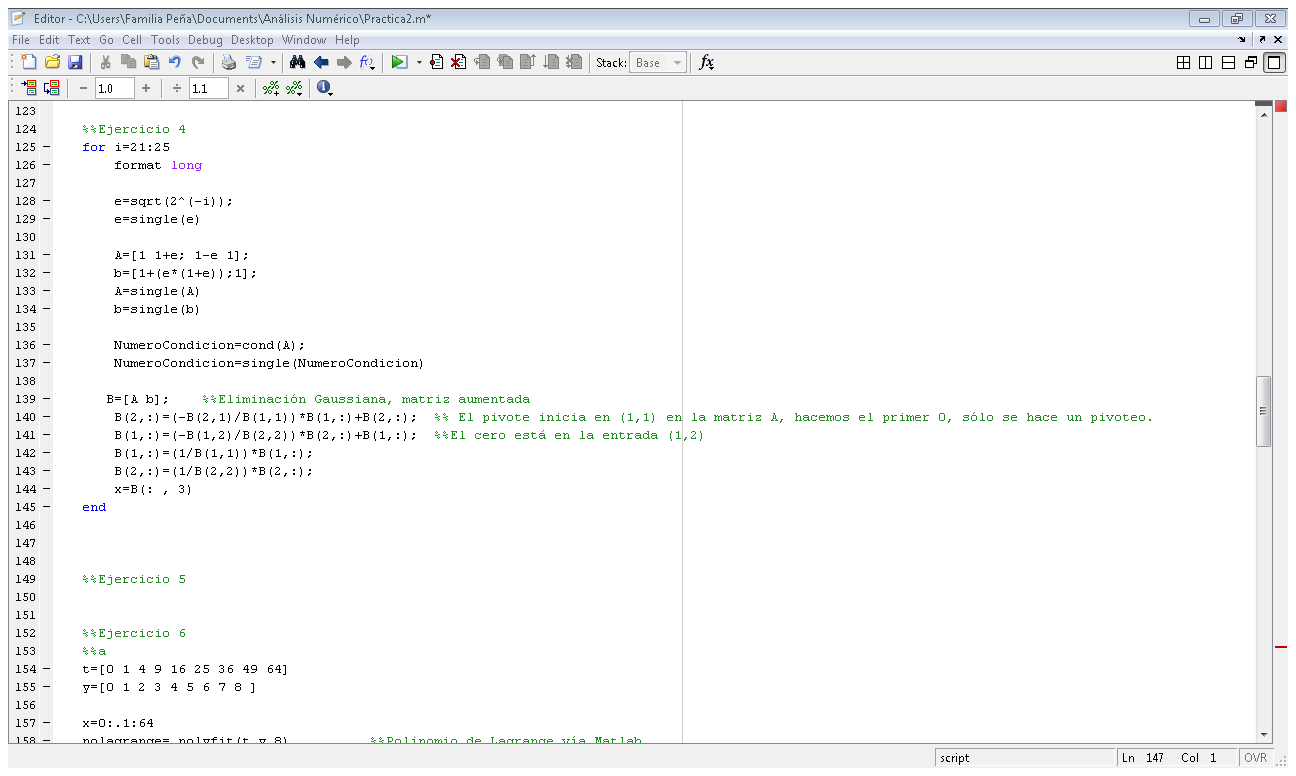


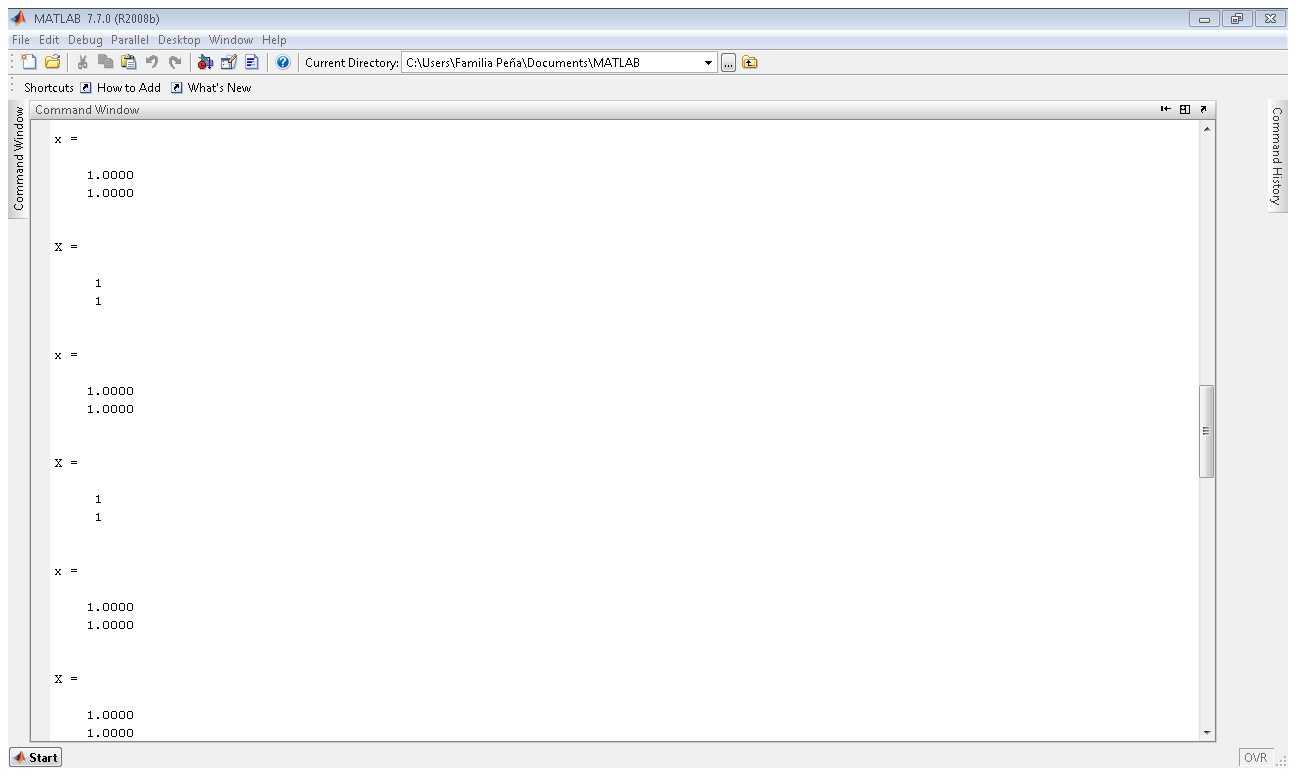
Ahora, cuando decrece, la solución se ve afectada por la pérdida de información, pues el método utiliza el primer elemento como pivote, el cual es y se va acarreando errores.

Como la computadora utiliza un sistema de punto flotante, tiene una cantidad finita de dígitos para guardar y siendo con muchos dígitos, ya sea muy grande o muy pequeño, en este caso pequeño, se pierden cifras significativas. Aunado a esto, el método utiliza la división del elemento pivote, lo que conlleva a mayores errores debido a ésta operación.

Notamos que para k=6 , la solución x es parecida a la matriz sugerida en el planteamiento del problema. Sin embargo, conforme k aumenta, es decir, decrece, la solución x cambia mucho a su antecesor y se aleja bastante de la matriz exacta que queríamos obtener x= .

b) Repetir la parte a) usando eliminación Gaussiana sin pivoteo, pero en esta ocasión usar una iteración de refinamiento para mejorar la solución calculando el residual en la misma precisión que el resto de los cálculos. ¿Cómo se comporta la precisión de la solución cuando decrece?

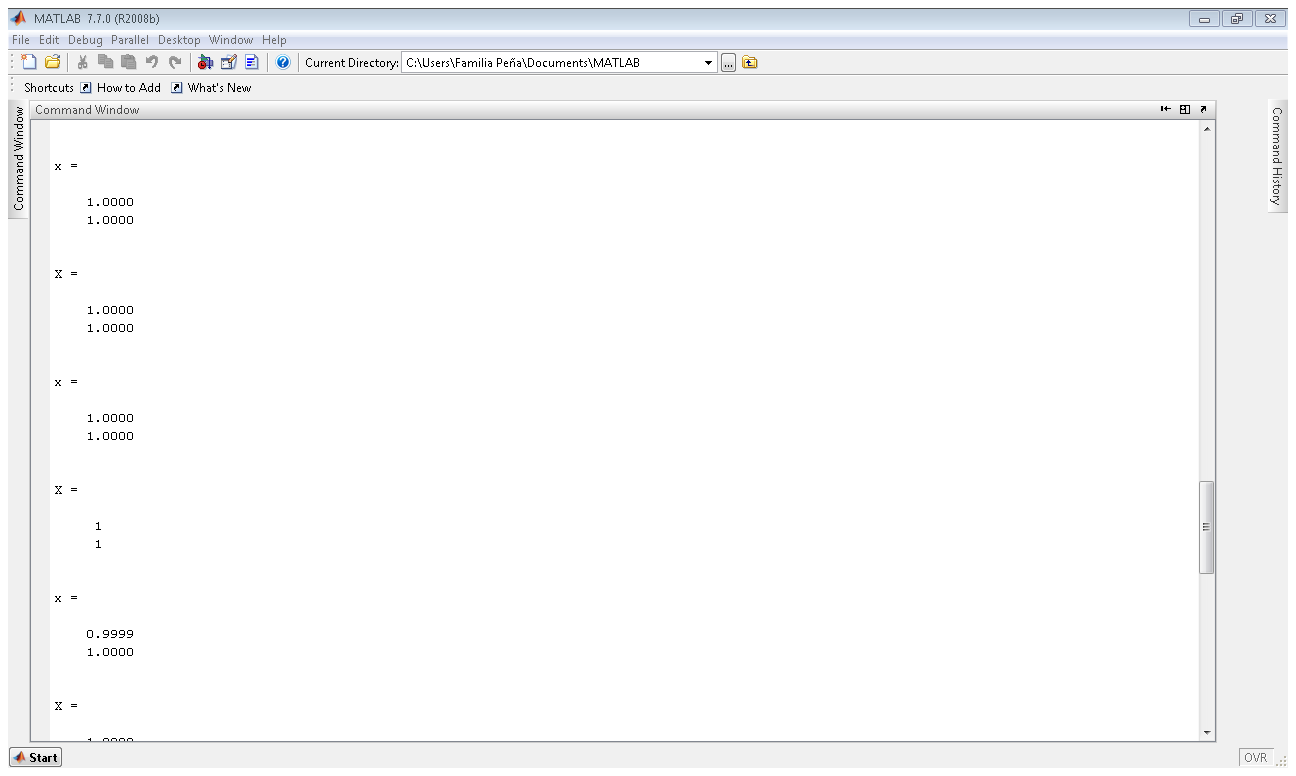




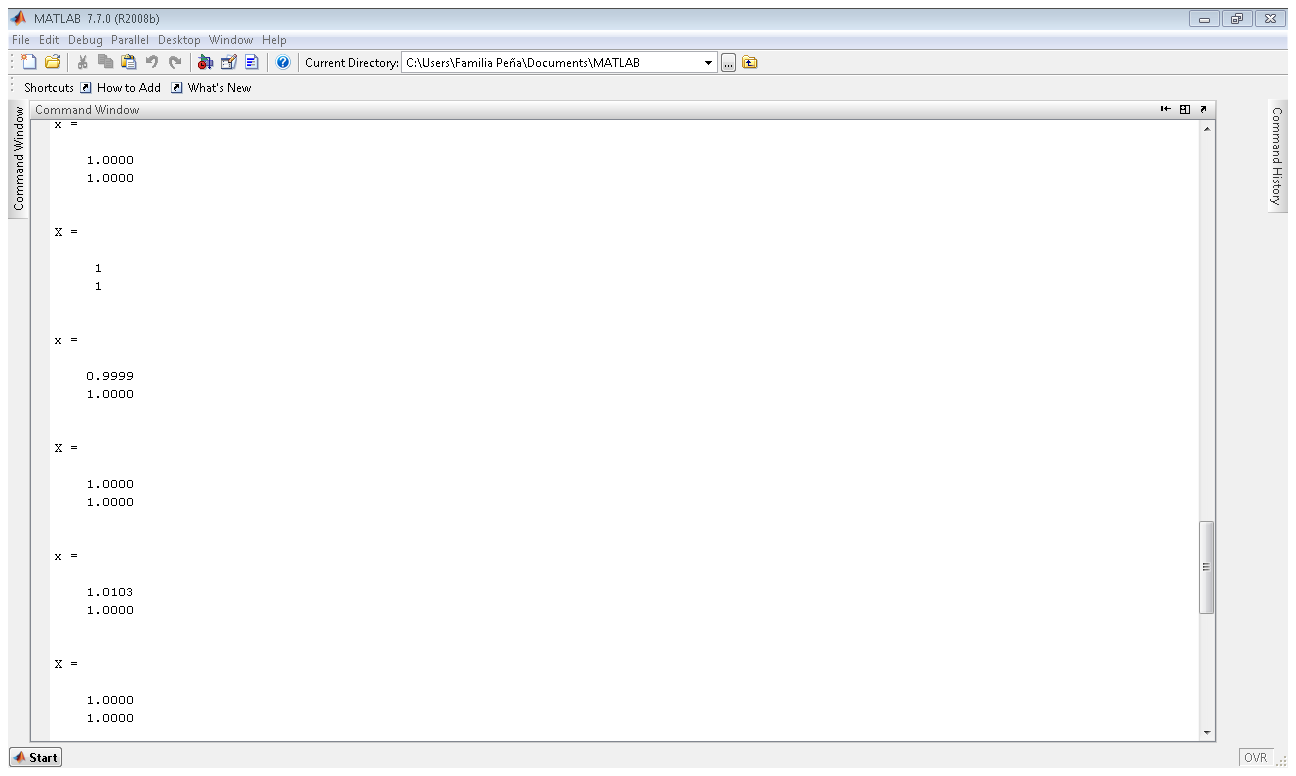
k=1

k=3

k=2



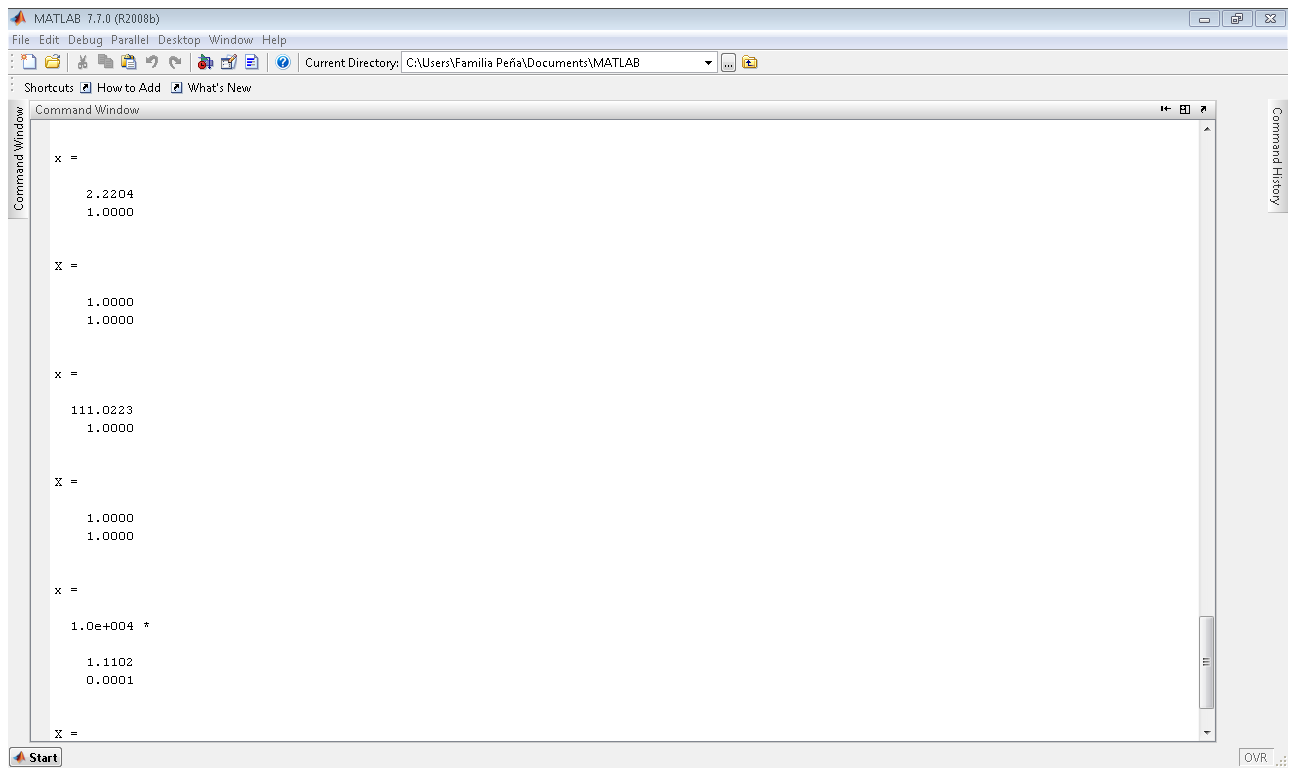
k=4



k=7

k=6

k=5



El refinamiento que se logra por este método, compensa el error producido por el cálculo del vector solución x. Cuando la k aumenta y por consiguiente, decrece, notamos que para x si hay cambios, sin embargo para la solución mejorada X notamos que la precisión de la solución permanece constante (1,1).

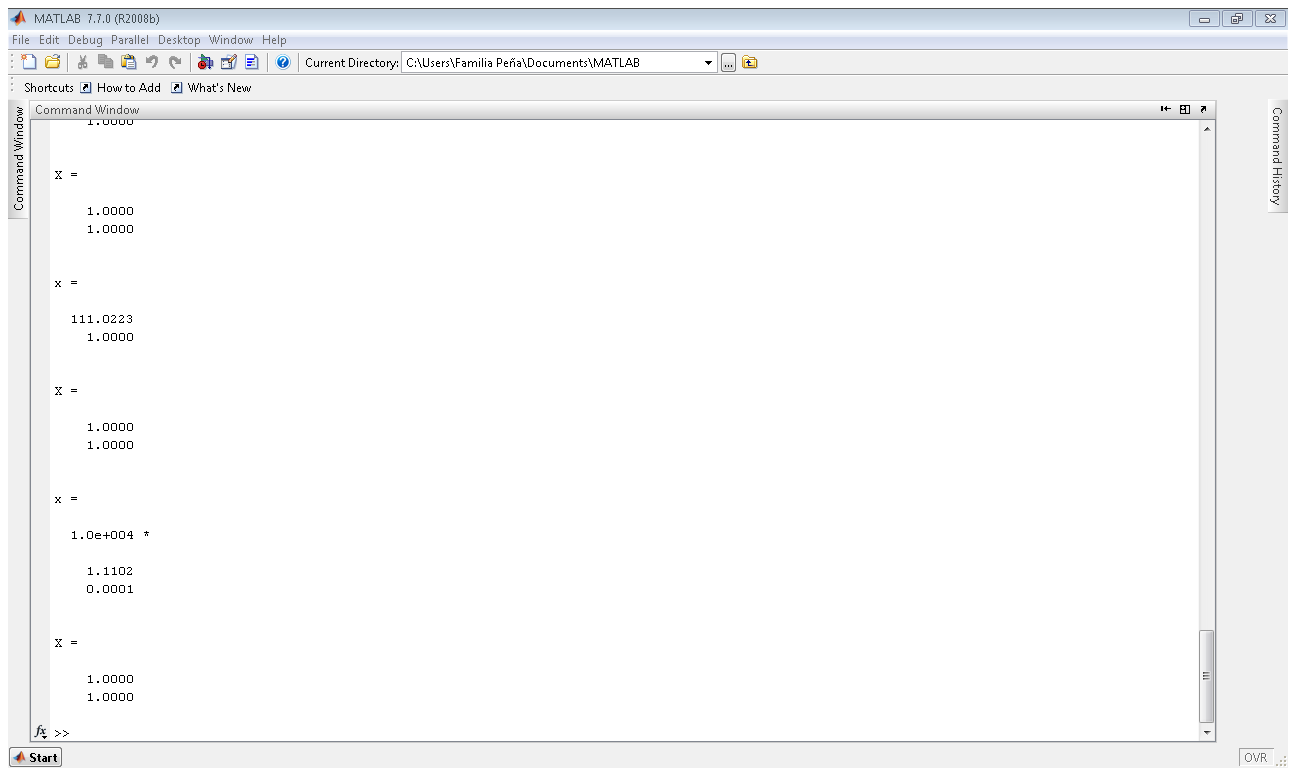
Este es un ejemplo de un problema mal condicionado, pues a la hora de meter los valores, la computadora puede tener problemas de precisión por el , el cual utiliza en éste inciso, un algoritmo estable, funciona, ya que la solución coincide con la exacta.

Entonces para la eliminación Gaussiana sin pivoteo y utilizando refinamiento, la solución numérica coincide con la exacta y es x= independiente del valor de .

k=10

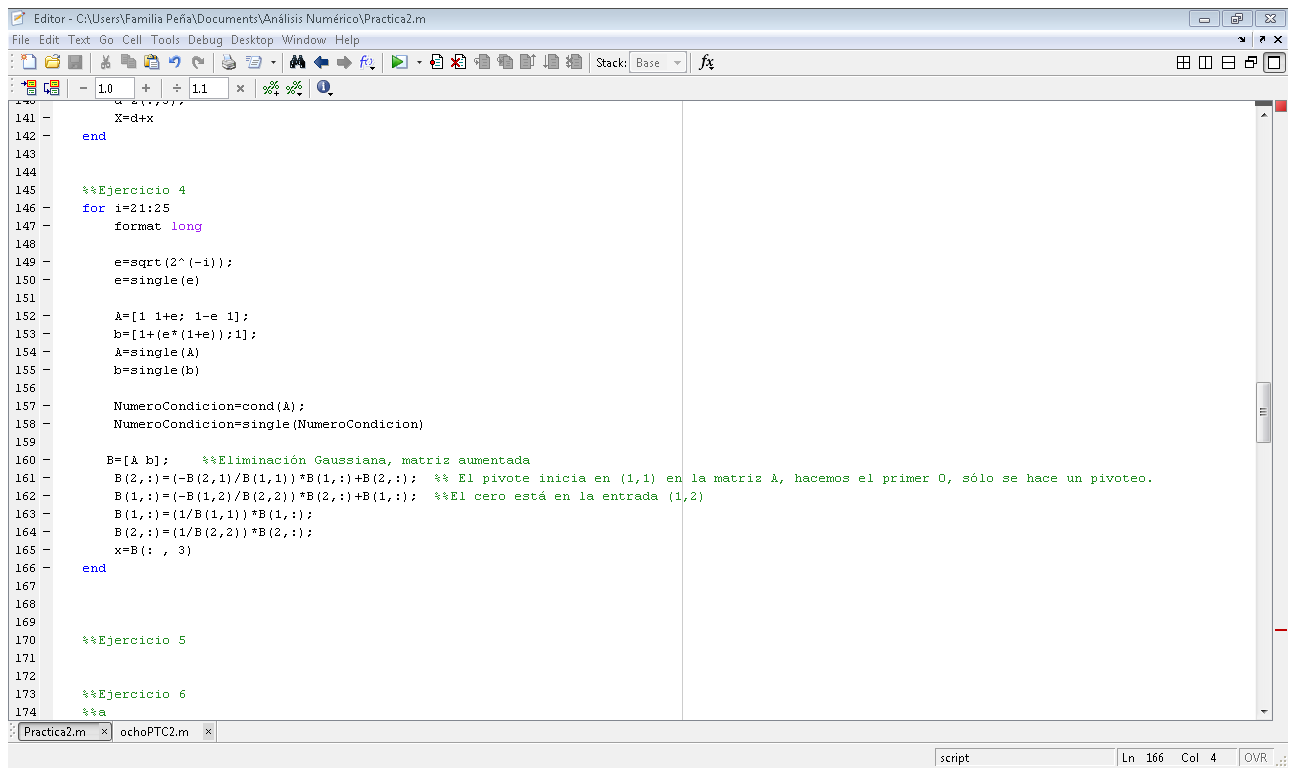
k=9

k=8

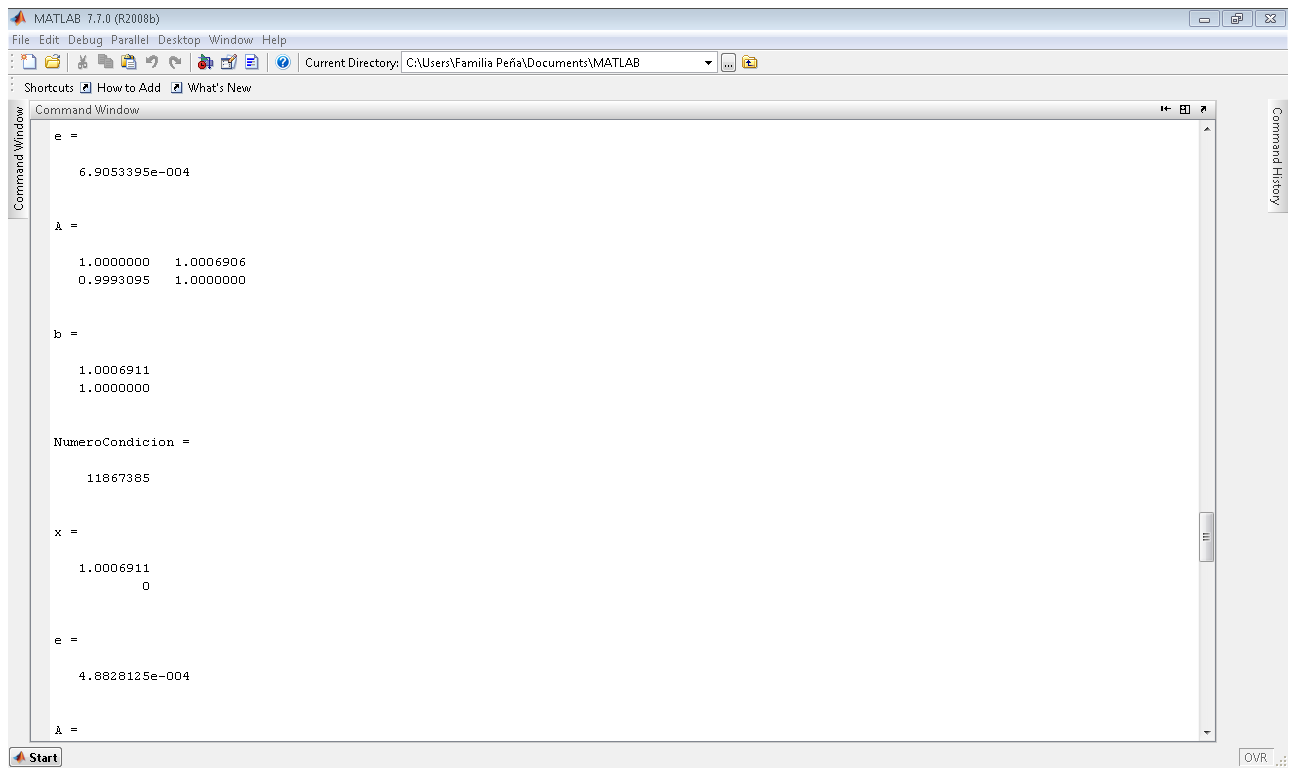


4. Considere el sistema lineal:

Donde es un parámetro a especificar. La solución exacta es x= . Usar una rutina de eliminación Gaussiana para resolver el sistema. Experimentar con diferentes valores de , especialmente con aquellos cercanos a . Para cada valor de calcula el número de condición de la matriz.



Para el formato simple tenemos que =, entonces el ciclo se hará alrededor de éste número.



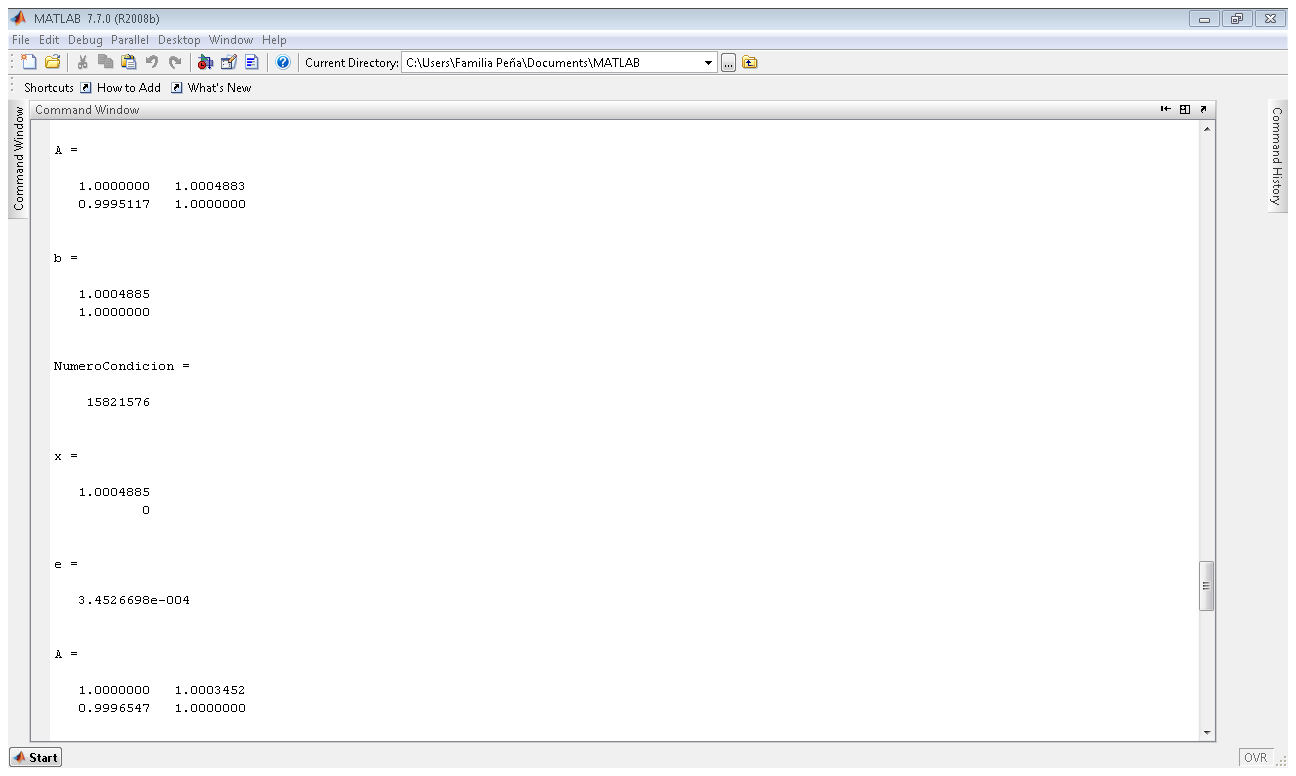
Como tenemos que la potencia va aumentando, el va decreciendo, éste número si llega a ser lo suficientemente pequeño (aunque no llega a ser el ).

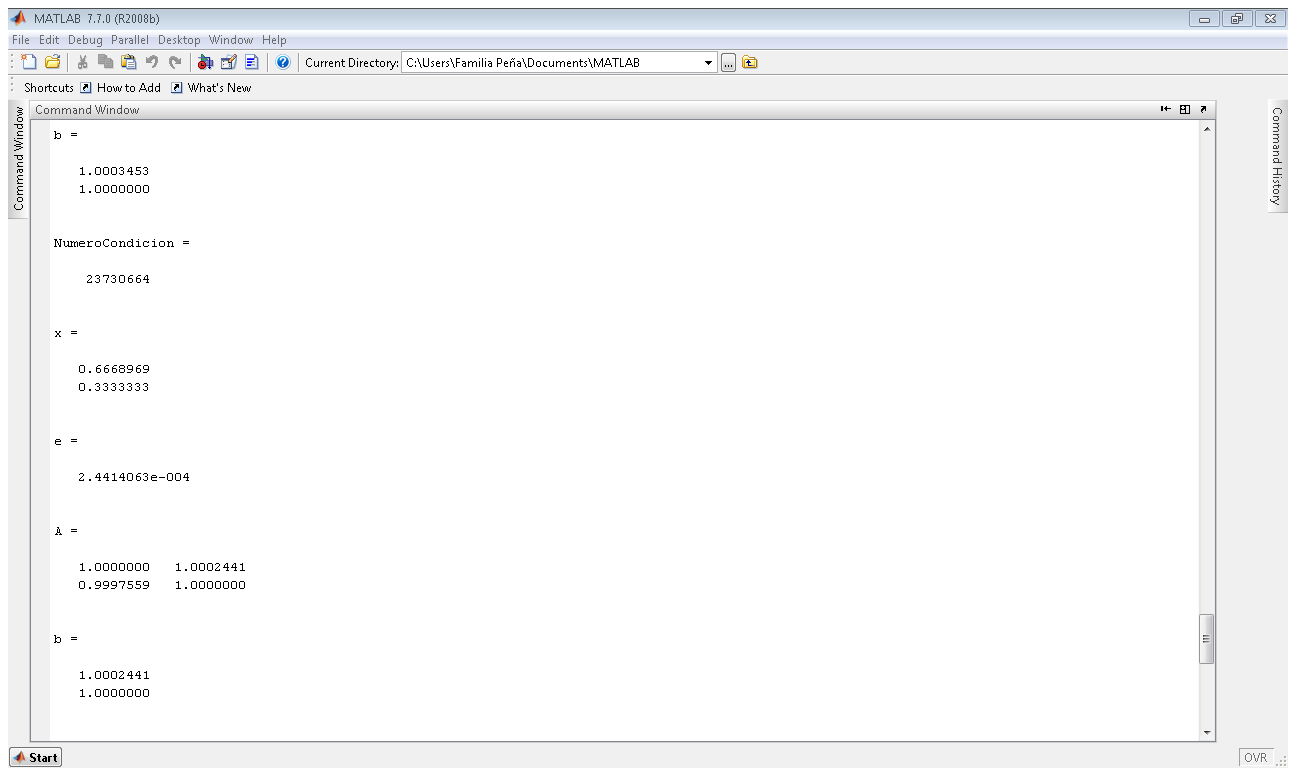
En consecuencia, provoca errores de redondeo y pérdidas significativas debido a las operaciones que desarrolla para el cálculo del resultado, lo cual se ve reflejado en el número de condición, que en la última iteración resulta ser infinito.

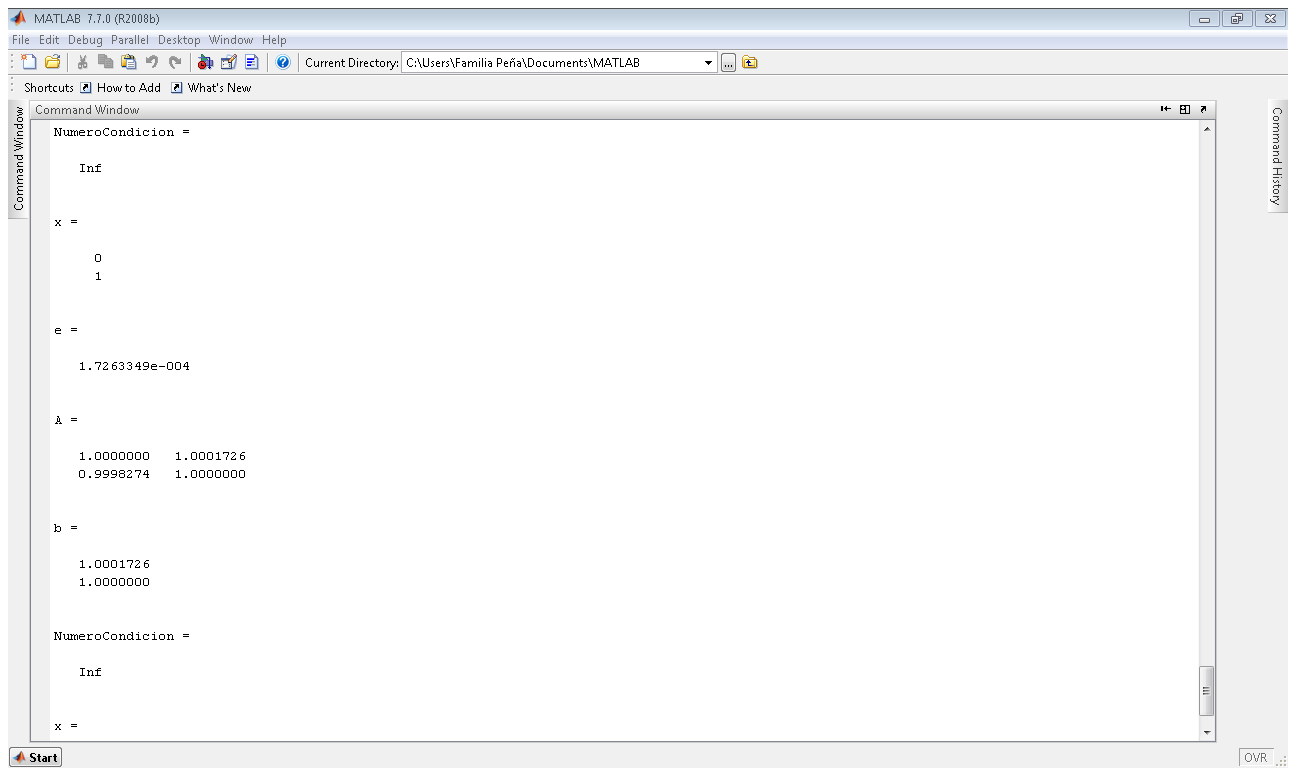
Observando el vector solución x para las primeras dos iteraciones, parece comportarse bien pues =0.0006905339, el cual ya es muy pequeño y la computadora lo redondea a cero. Entonces en dichas iteraciones la solución numérica y la solución exacta x= parecen ser las mismas.

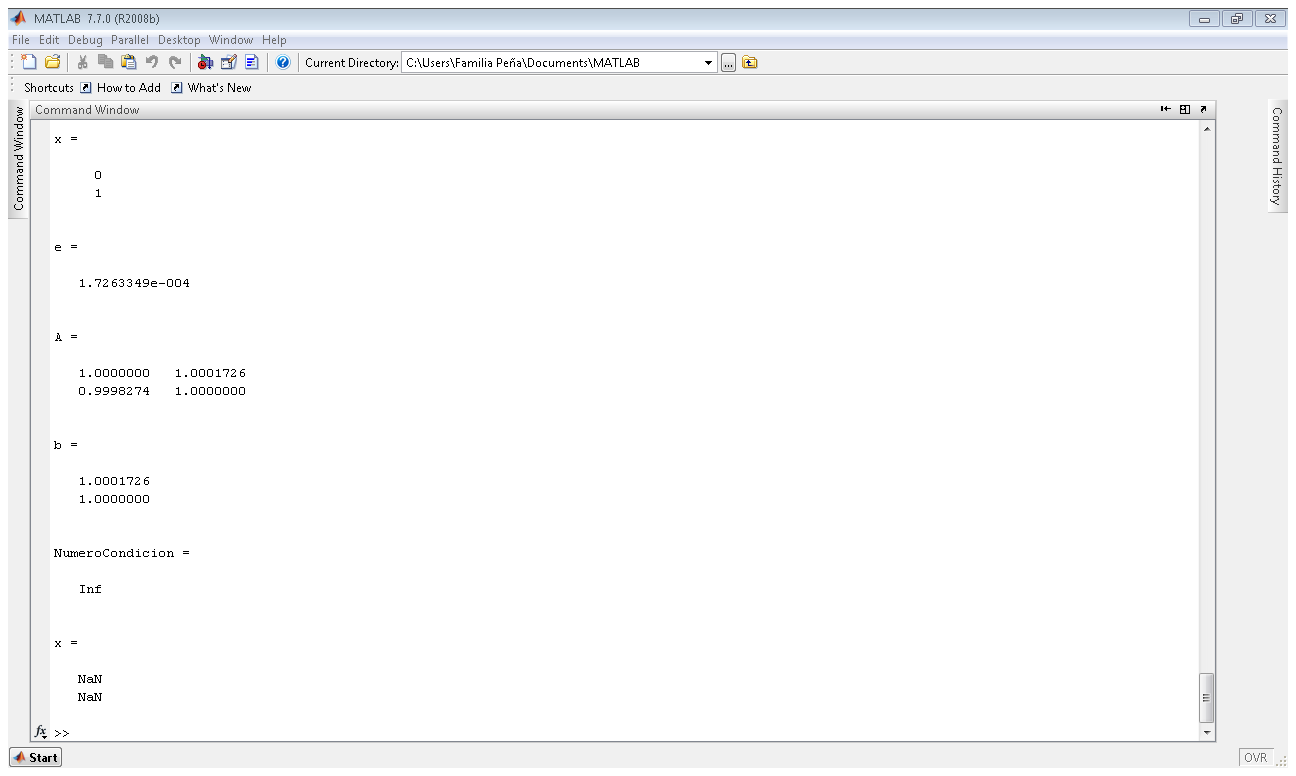
Sin embargo, a partir de la tercera iteración, la solución en aritmética del punto flotante es diferente a la solución con aritmética exacta, que como ya habíamos mencionado anteriormente descrito, debido a pérdidas significativas y a operaciones.

Para la última iteración que es , donde el es más pequeño, notamos que el vector solución x ya acarrea más error pues empieza a ser divisiones entre cero debido al redondeo por ser tan pequeño mostrando error y es x=(NaN, NaN) la solución, la cual es sumamente diferente a la matriz solución exacta.

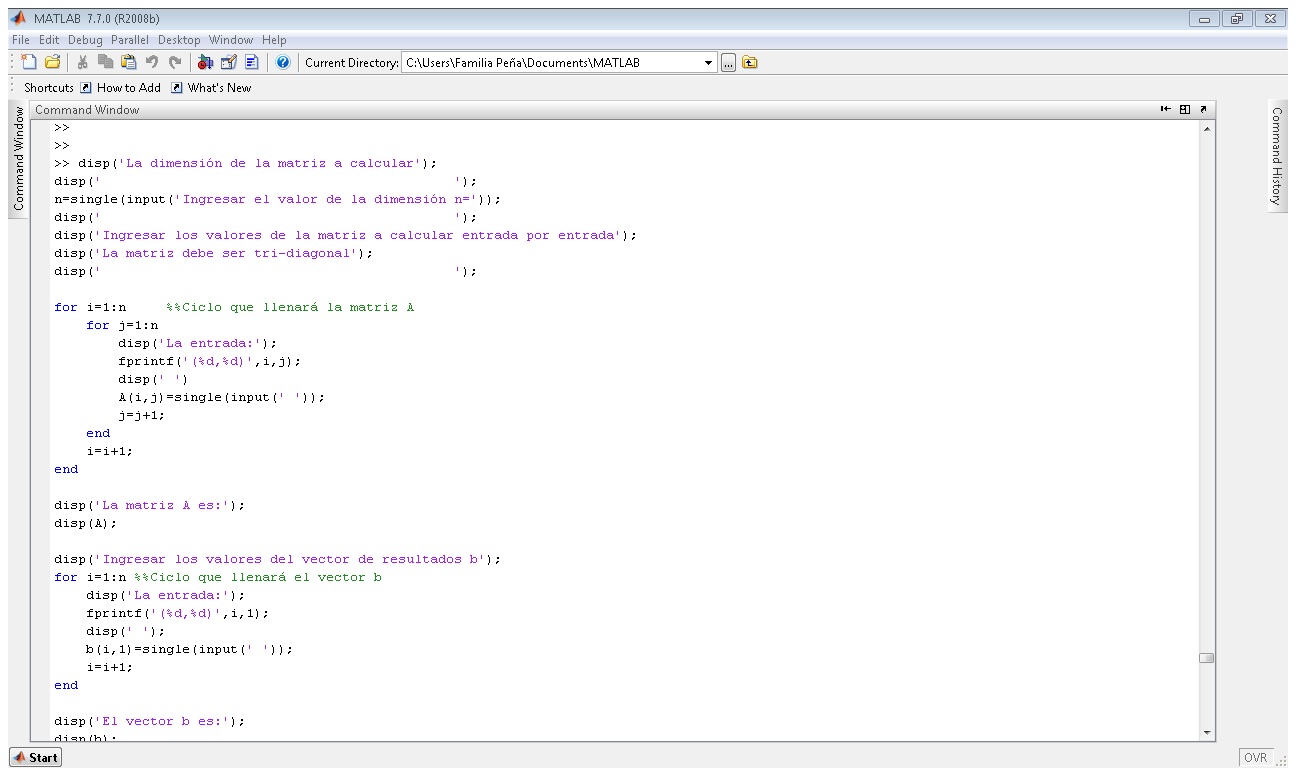


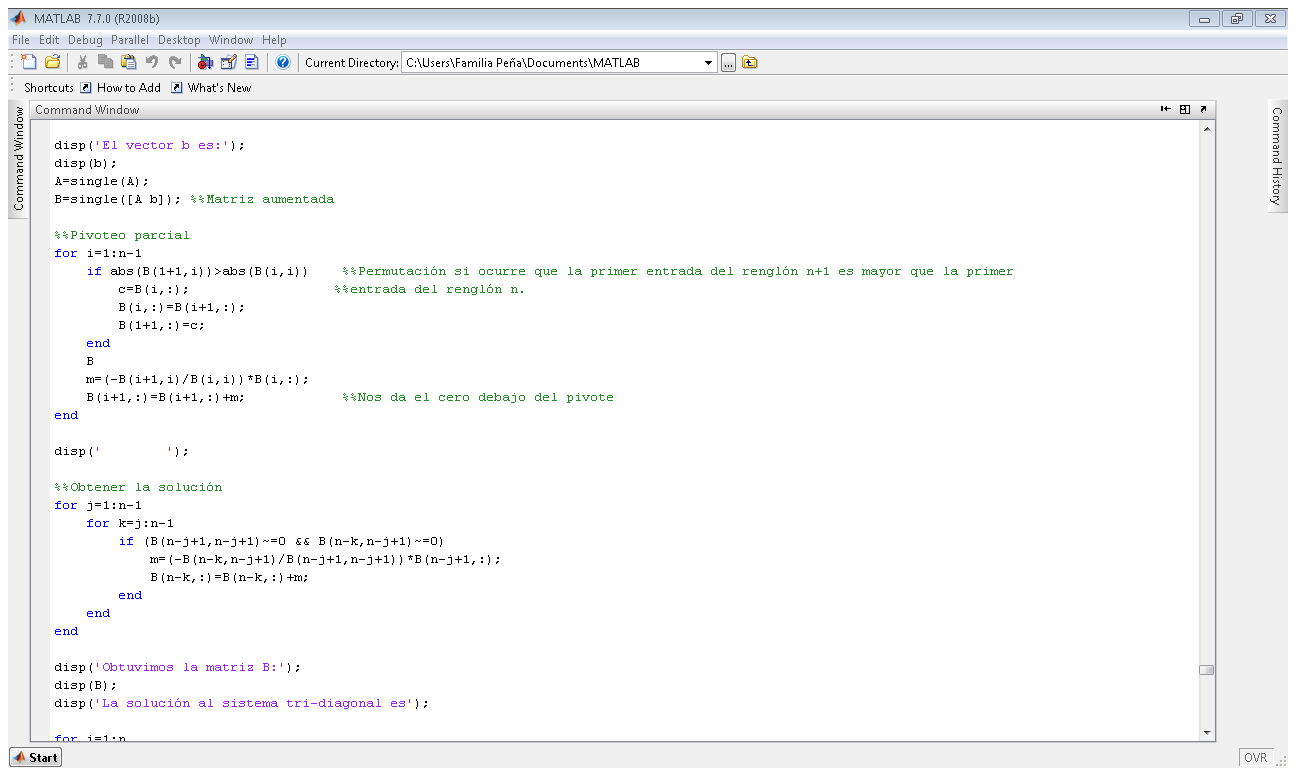


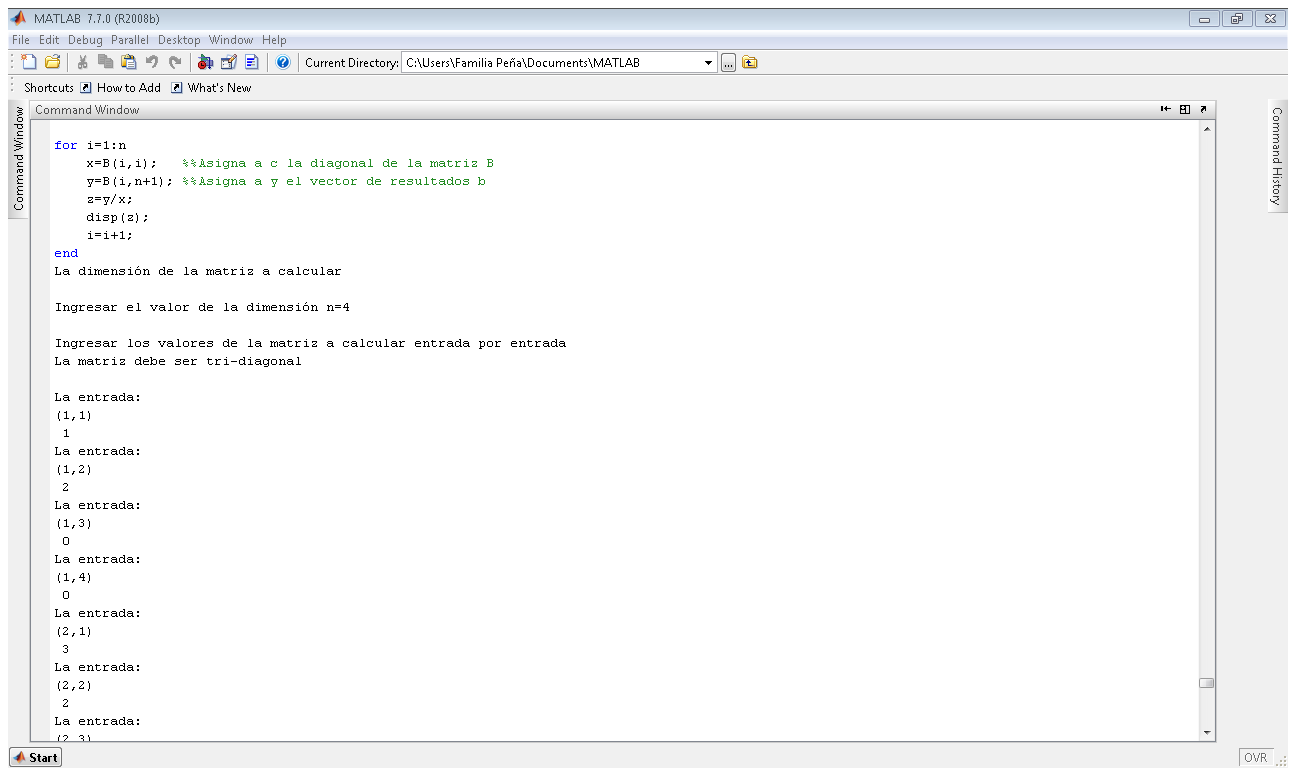


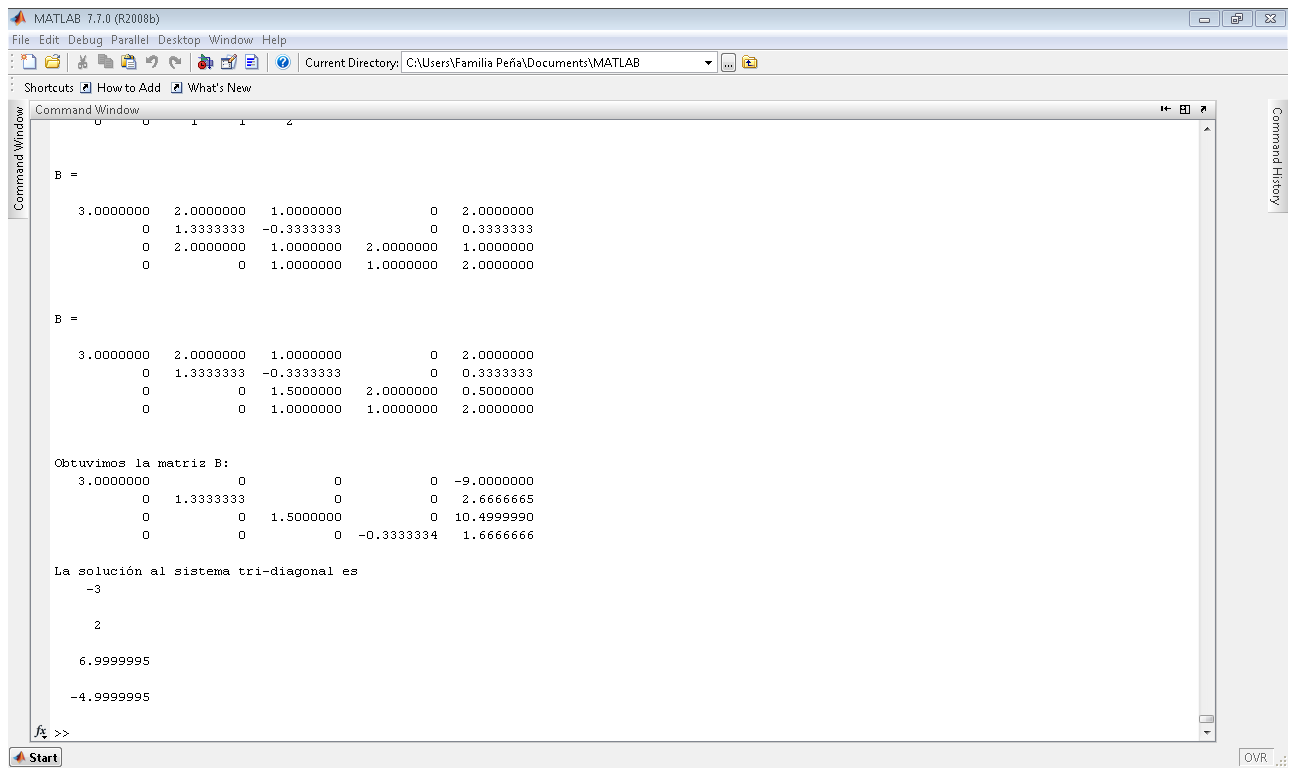
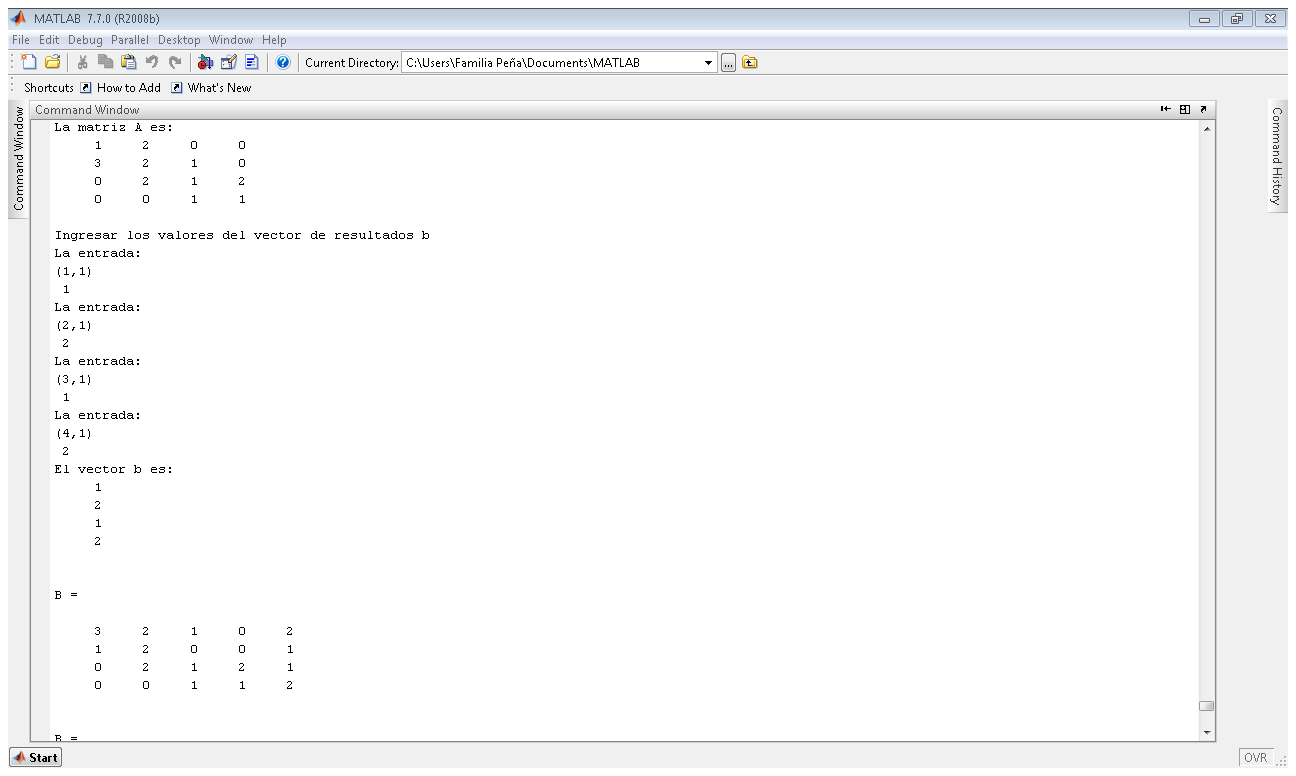


5. Escribir una rutina que resuelva sistemas de ecuaciones tri-diagonales usando pivoteo parcial. Sugerencia ver el algoritmo en la sección 2.5.3 del libro de Michael Heath.

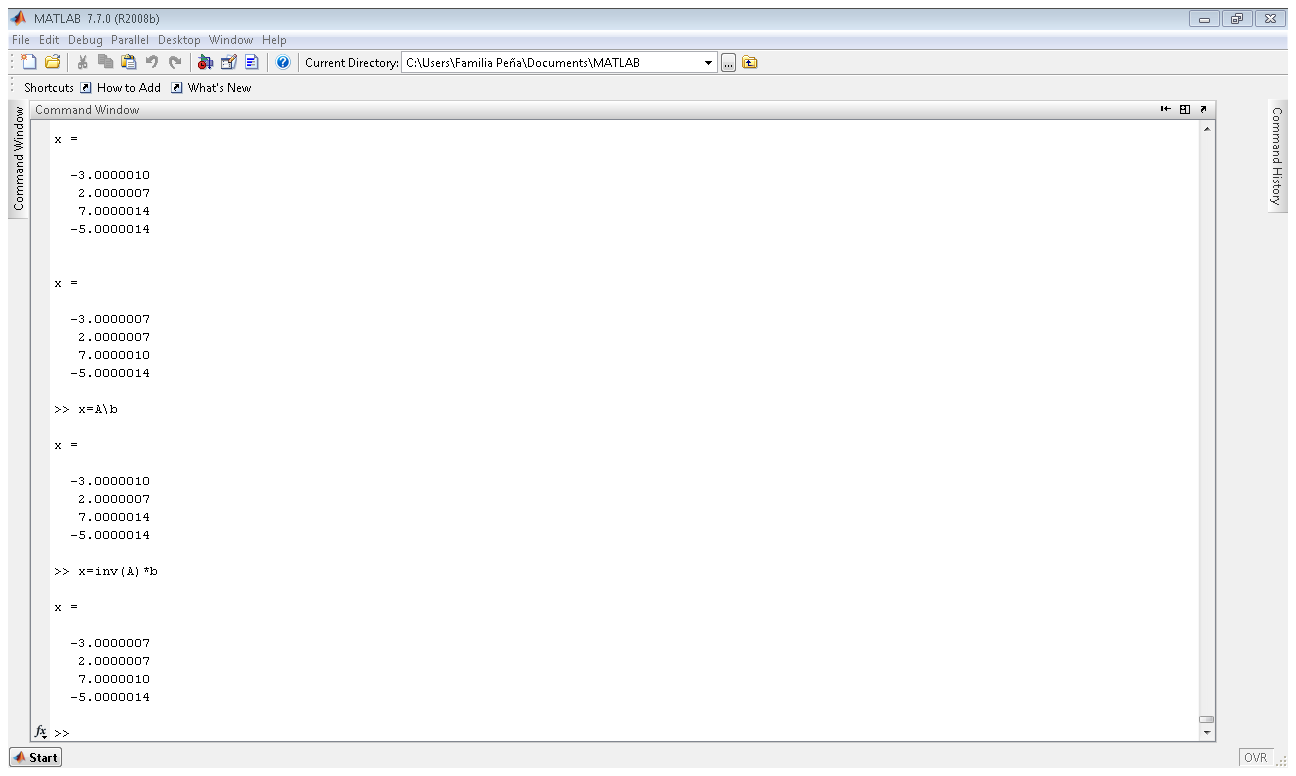
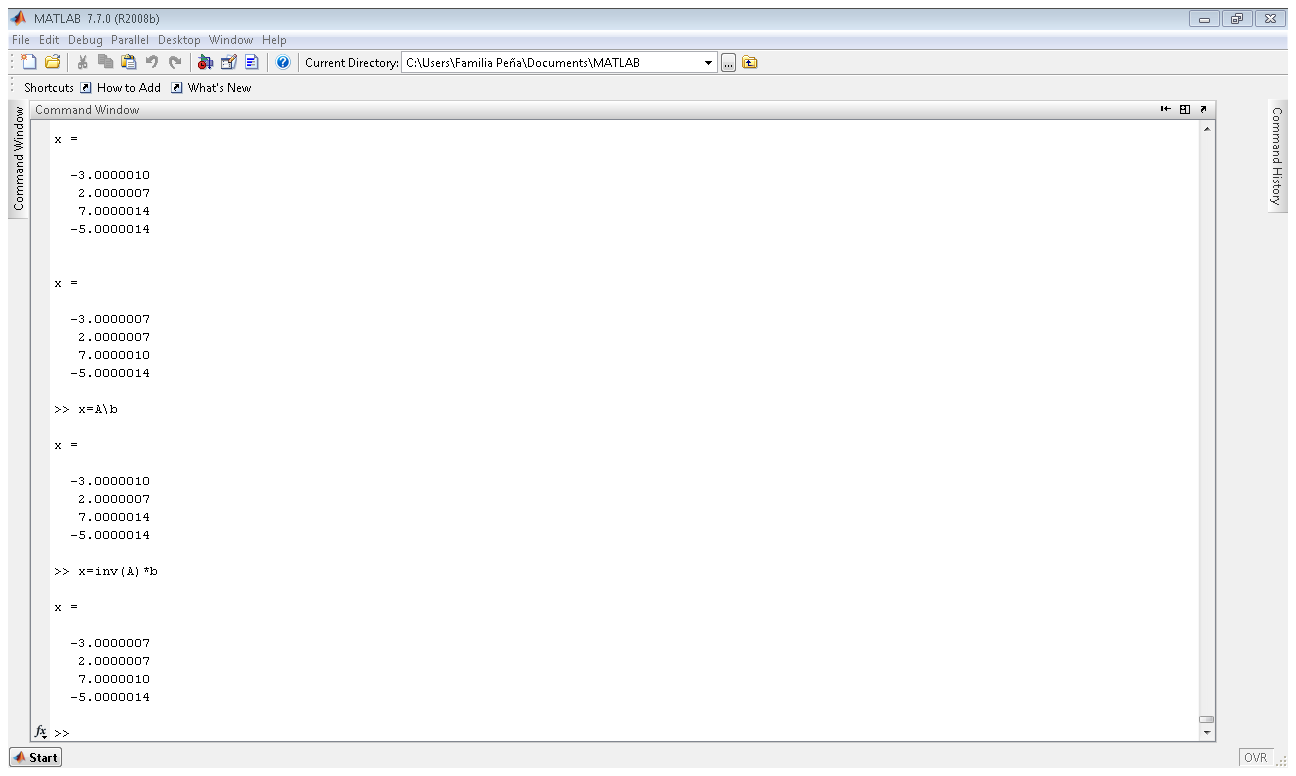








Ahora veamos cuál es el resultado con los otros métodos que hemos utilizado y así comparar.



En un principio notamos que las soluciones entre eliminación Gaussiana utilizando pivoteo parcial y los otros dos, son bastante aproximados. Observamos que con eliminación Gaussiana con pivoteo parcial obtenemos una mayor precisión pues se evita la propagación del error, sin embargo como el algoritmo consiste en tener el pivote más grande, requiere de hacer operaciones con las matrices de permutación, las cuales generan mayor costo por necesitar más tiempo en comparar.

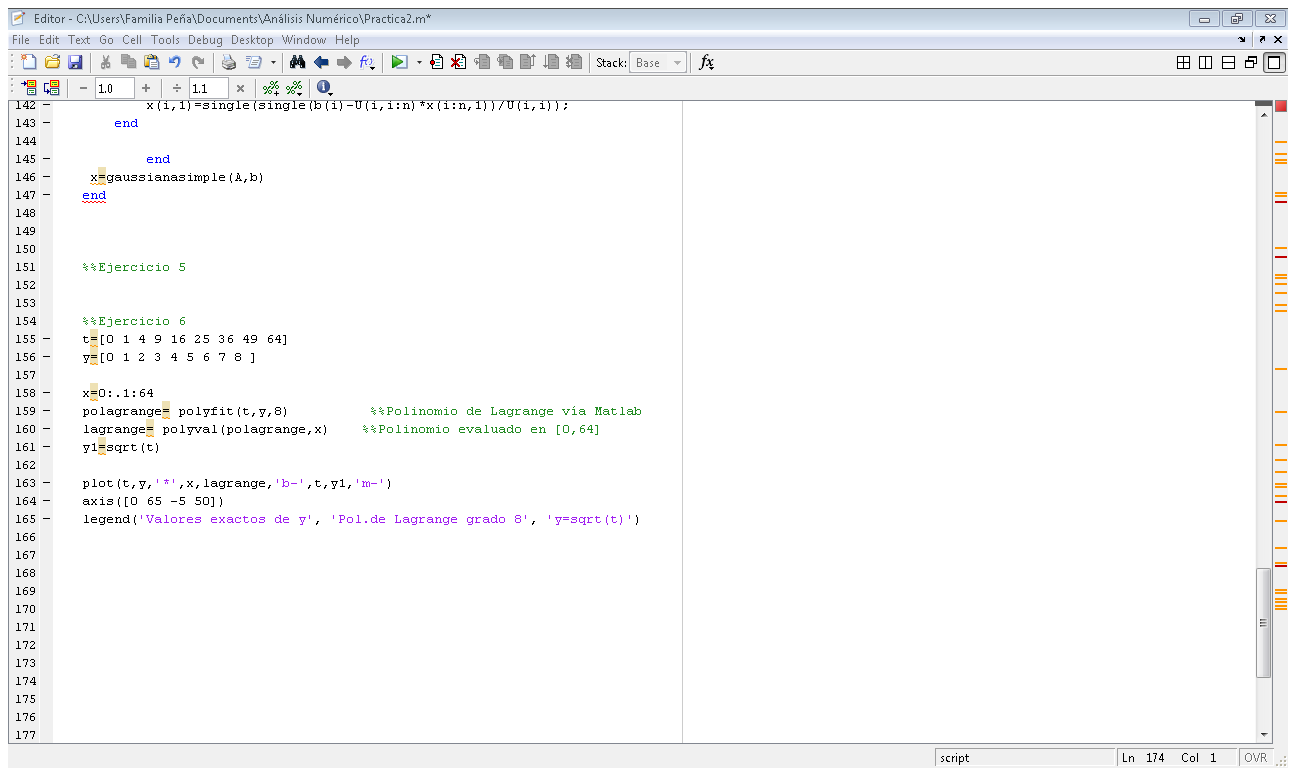
6. Interpolating the data points

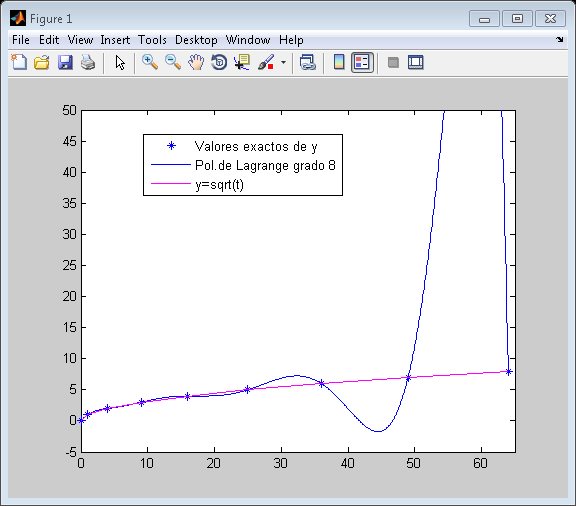
t 0 1 4 9 16 25 36 49 64

y 0 1 2 3 4 5 6 7 8

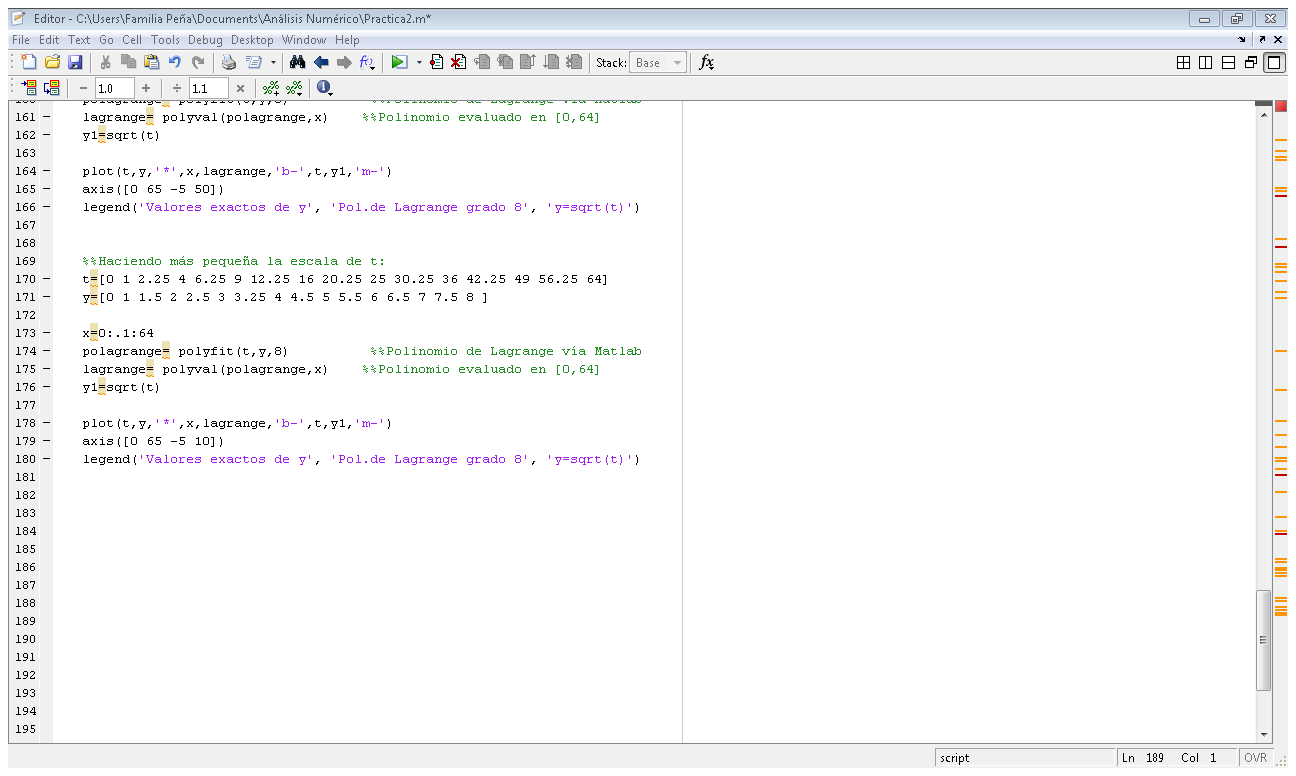
Should give an approximation to the square root function

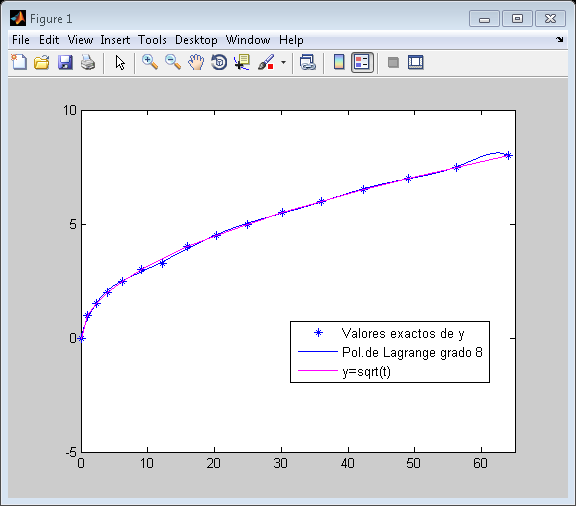
a) Compute the polynomial of degree eight that interpolates these nine data points. Plot the resulting polynomial as well as the corresponding values given by the biolt-in sqrt function over the domain [0;64]





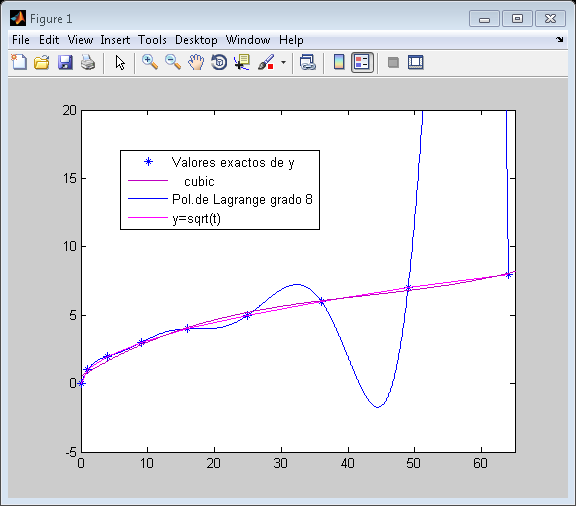
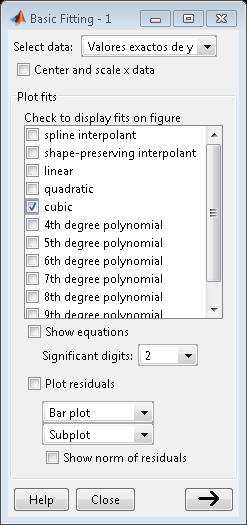
Notamos que el Polinomio de Lagrange se aproxima bien a la función raíz para los primeros cuatro valores, después de esto se aleja bastante. Esto se podría mejorar si la escala de t, y se hace más pequeña.





Para la nueva escala, notamos que es se aproxima bastante ben el polinomio a la función sqrt, a pesar que sigue siendo de grado 8. Sólo en el último valor difieren un poco.

b) Use a cubic spline routine to interpolate the same data and again plot the resulting curve along with the built.in sqrt function.



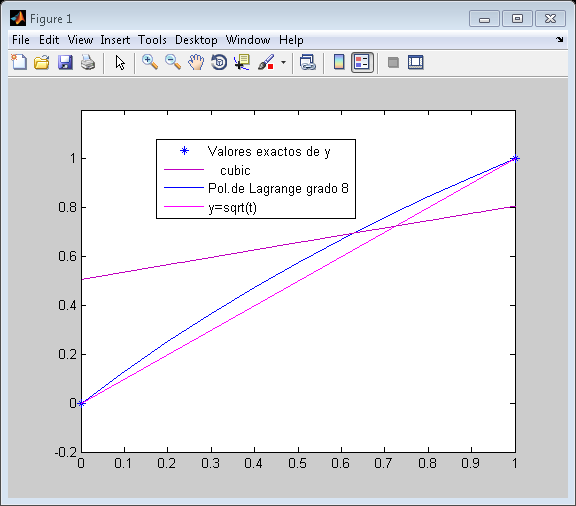
Notamos que utilizando una función cúbica, se adapta bien a la función y=sqrt(t), pues se aproxima bastante en el dominio establecido. Solamente que en un inicio es cuando se separa más a la función sqrt(t), para después aproximarse más cuando t crece. Incluso, podría pensarse que se aproxima mejor que el polinomio de Lagrange de grado 8.

c) Which of the two interpolant is more accurate over most of the domain?

De manera empírica, notamos que en el inciso a) refinando nuestro intervalo, se aproxima para los primeros valores, sin embargo para los valores t, y dados difiere bastante. Lo que para la función de splines cúbicos establecida por Matlab, se aproxima bien a la función sqrt(t) en todo el dominio [0,64]. En conclusión, establecemos que la función de splines cúbicos es más exacta, es decir se aproxima más a la sqrt(t) que Polinomio de Lagrange de grado 8 en éste dominio.

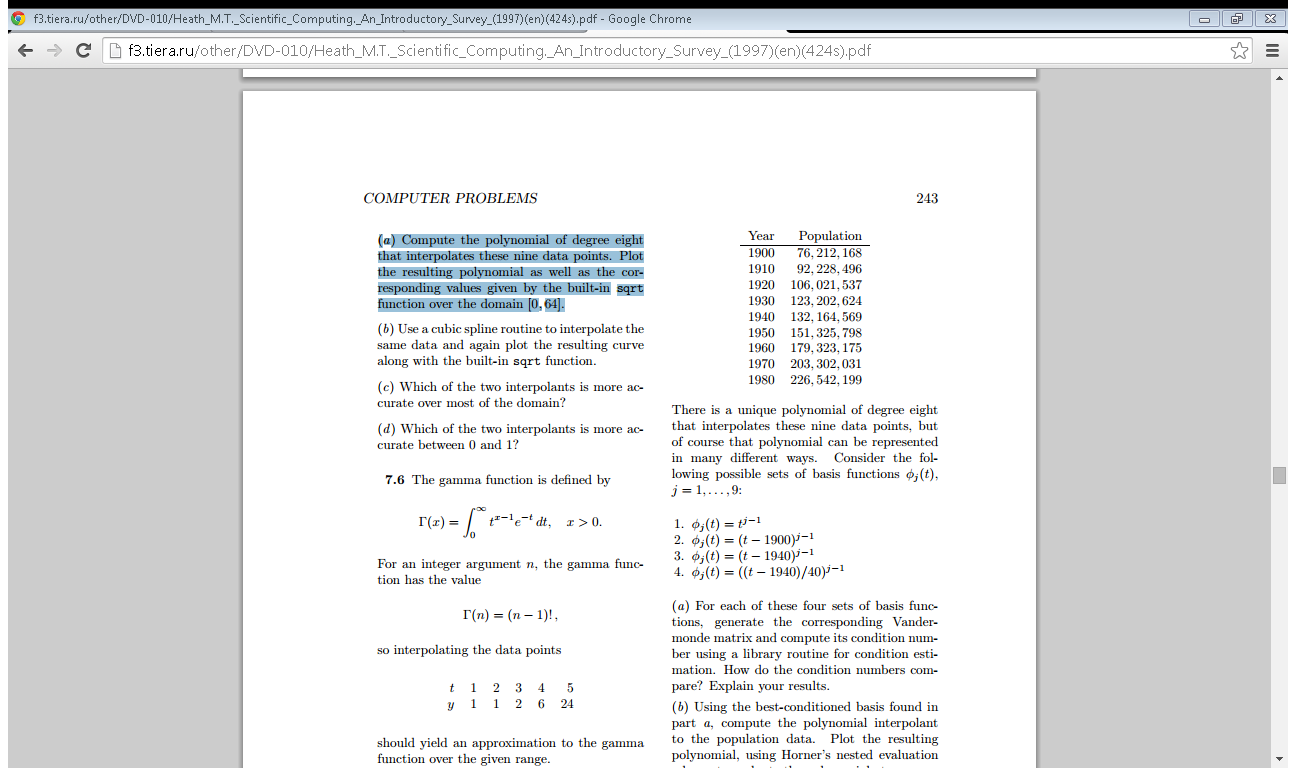
De manera formal, la distancia entre cada función se calculará con la norma dos vía Matlab. Notamos que la distancia entre el polinomio de Lagrange y la función sqrt(t) es 709.4799, mientras que la distancia entre el spline cúbico y la función es 0.8595. El polinomio no es una buena aproximación pues además que no se encuentra tan cerca y es una función oscilatoria, por lo tanto, como habíamos notado anteriormente, el spline cúbico es mejor aproximación.

d) Which of two inteerpolants is more accurate between 0 and 1



Notamos que el Polinomio de Lagrange de grado 8 se aproxima mejor a la función y=sqrt(t) en el dominio (0,1), mejor que el spline cúbico (natural, esto es debido a que tenemos la misma cantidad de nodos que de datos).

7. Ejercicio 7.7 (pag. 337 edición Impresa 2005) del libro de Michael Heath. (NOTA: En el archivo PDF que les envié este ejercicio es el 7.6, pero de la pag. 243, no la numeración del archivo PDF si no la página)

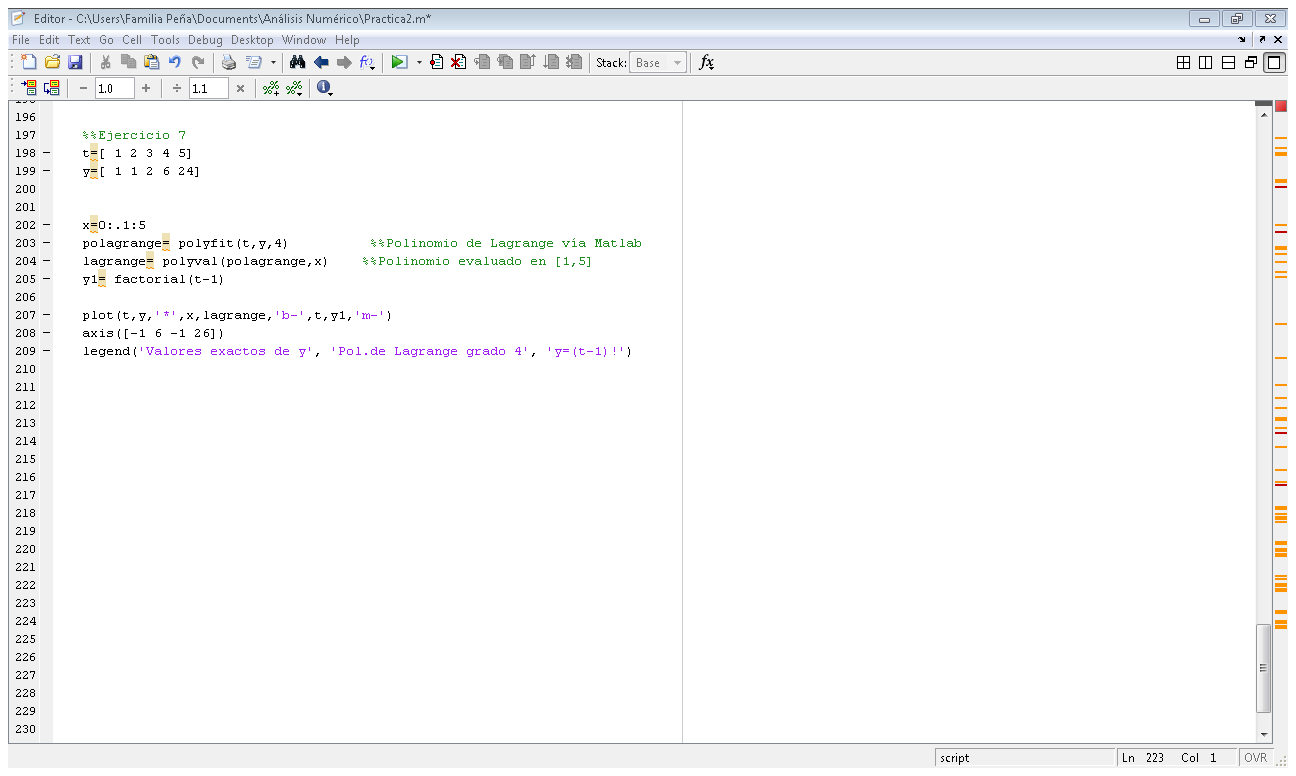


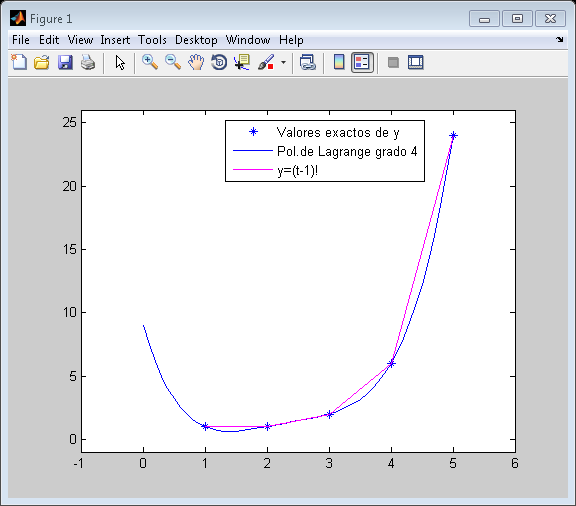
So interpolating the data points

t 1 2 3 4 5

y 1 1 2 6 24Should yield an approximation to the gamma function over the given range.

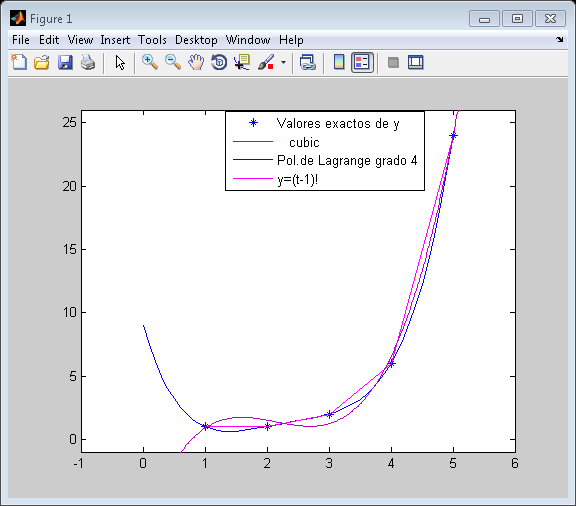
a) Compute the polynomial of degree four that interpolates these fives data points. Plot the resulting polynomial as well as the corresponding values given by the buit-in gamma function over the domain [1,5].





Notamos que en el subintervalo [2,3] coinciden, sin embargo para los demás subintervalos no se aproximan tanto, pareciera que el Polinomio de Lagrange de grado 4 está más suavizado que la función factorial.

b) Use a cubic spline routine to interpolate the same data and again plot the resulting curve along with the built-in gamma function.



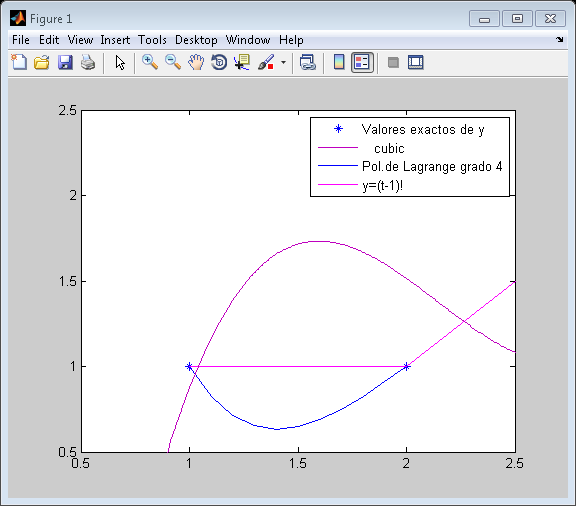
Observamos que para los primeros valores, la función cúbica no es tan aproximada a la función gamma, sin embargo, para valores más grandes, si se va ajustando mejor a dicha función, en el intervalo [1,5].

c) Which of the two interpolants is more accurate over most of the domain?

A diferencia del ejercicio 6, no tenemos una aproximación que resulte idónea. Así que si observamos la función empleada por Matlab para los splines cúbicos (naturales, pues similar al ej.6 tenemos la misma cantidad de nodos que de datos) se ajusta a la función gamma para valores mayores a 4 nadamás. Sin embargo notamos que el polinomio de Lagrange de grado 4 en el dominio [1,5] se aproxima bien a la función y=(t-1)!

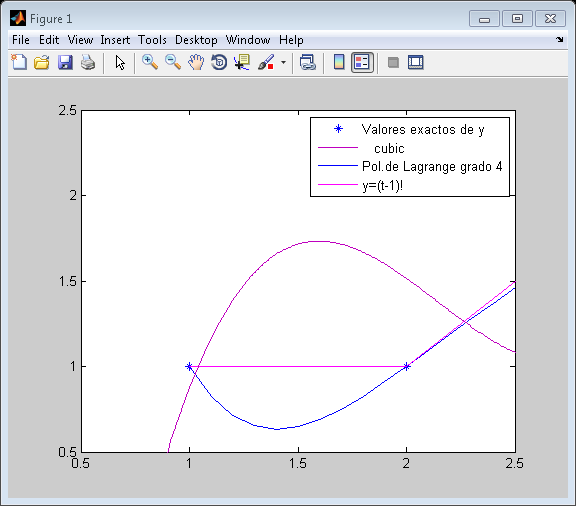
De manera formal, como no podemos observar a primera instancia, qué función se aproxima mejor, utilizamos la norma dos vía Matlab. La norma entre Lagrange y la función gamma es de 1.6528 y la norma entre el spline cúbico y la función es de 2.0634, por lo que, a diferencia de lo observado a simple vista, la interpolación más exacta sobre el dominio [1,5] es el spline cúbico.

d) Which of the two interpolant is more acurate between 1 and 2?



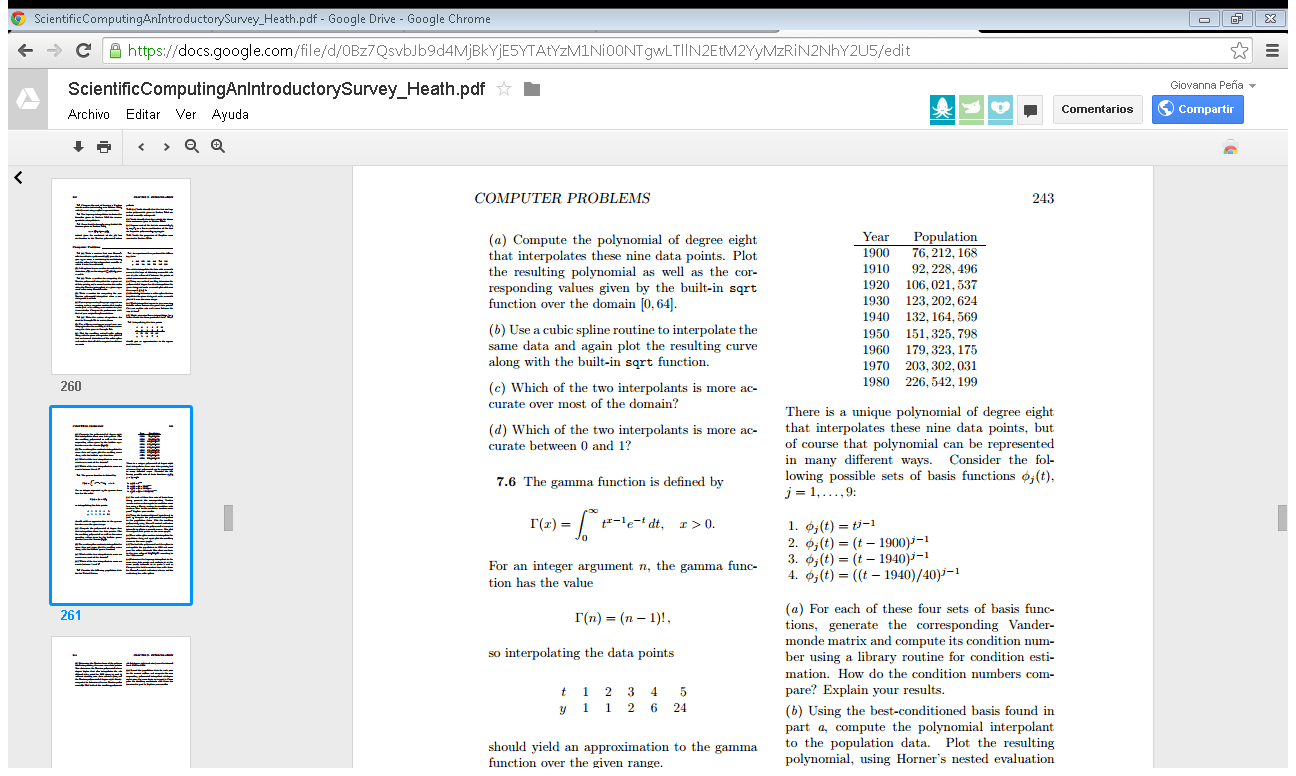
En este intervalo no notamos mucha contribución por parte del Spline cúbico en el dominio [1,2] pues no se aproxima mucho a la función Gamma. Entonces, podemos observar que la función que más se aproxima a la gamma es nuevamente el Polinomio de Lagrange de orden 4, que bien, aunque no es una aproximación tan exacta, es mejor que la cúbica.

Mejor analizaremos qué función se aproxima a la gamma más haciendo el intervalo ligeramente más grande.

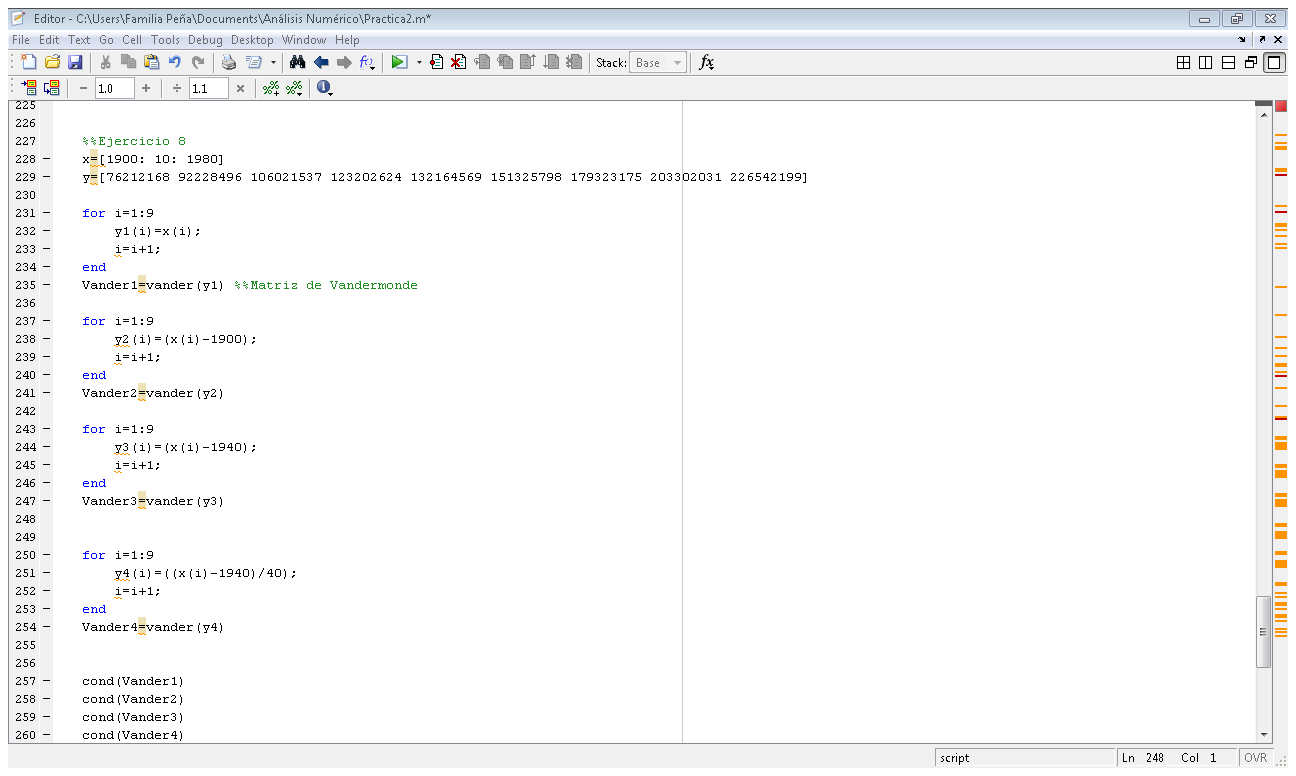


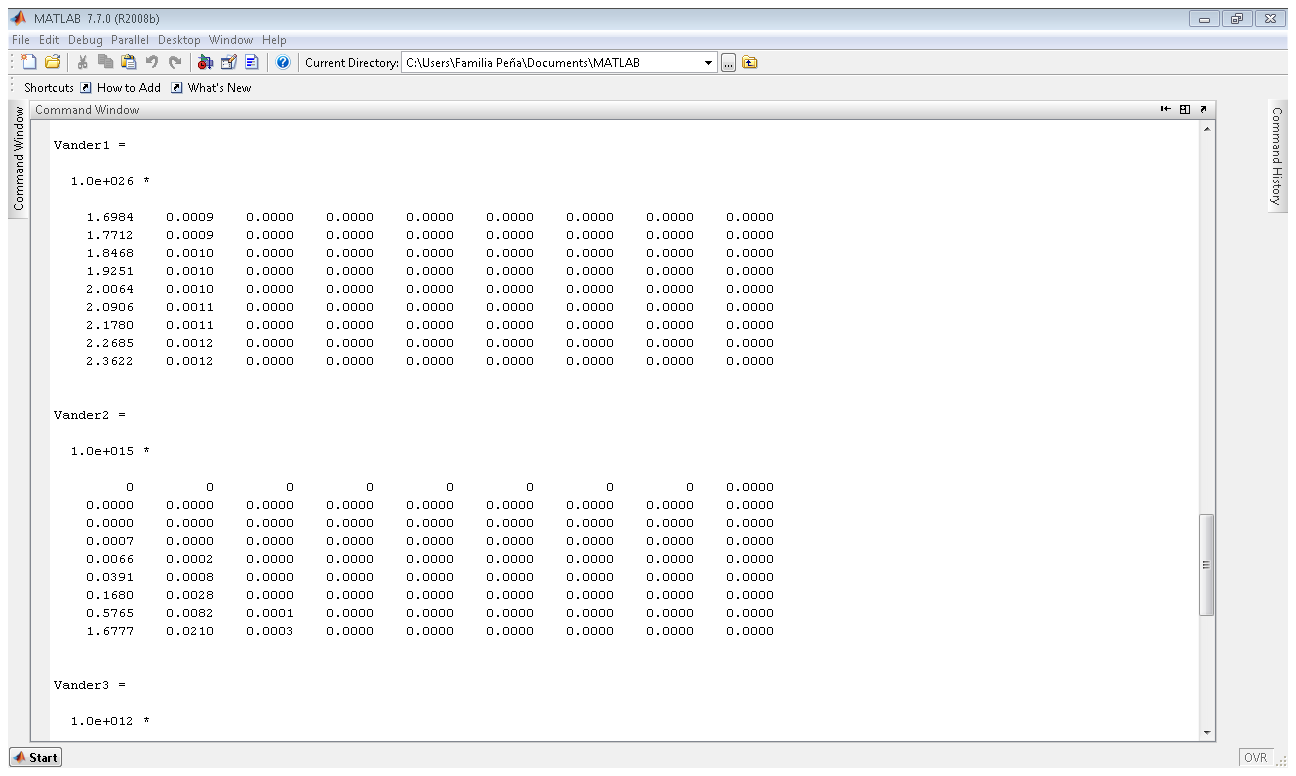
Ahora, notamos que aunque hicimos el intervalo más grande, la función que más se aproxima a la gamma es la del Polinomio de Lagrange de orden 4 en el dominio [1, 2.5].

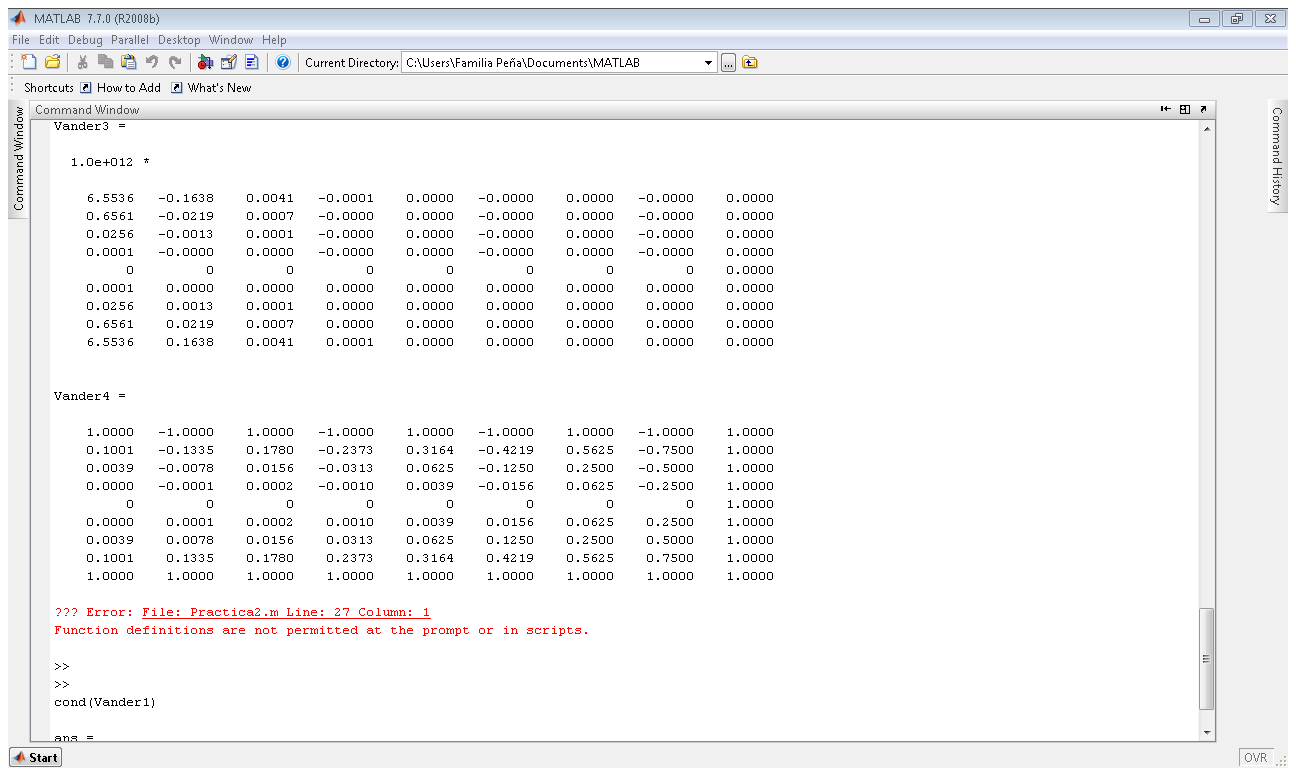
8. Consider the following population data for the United States:

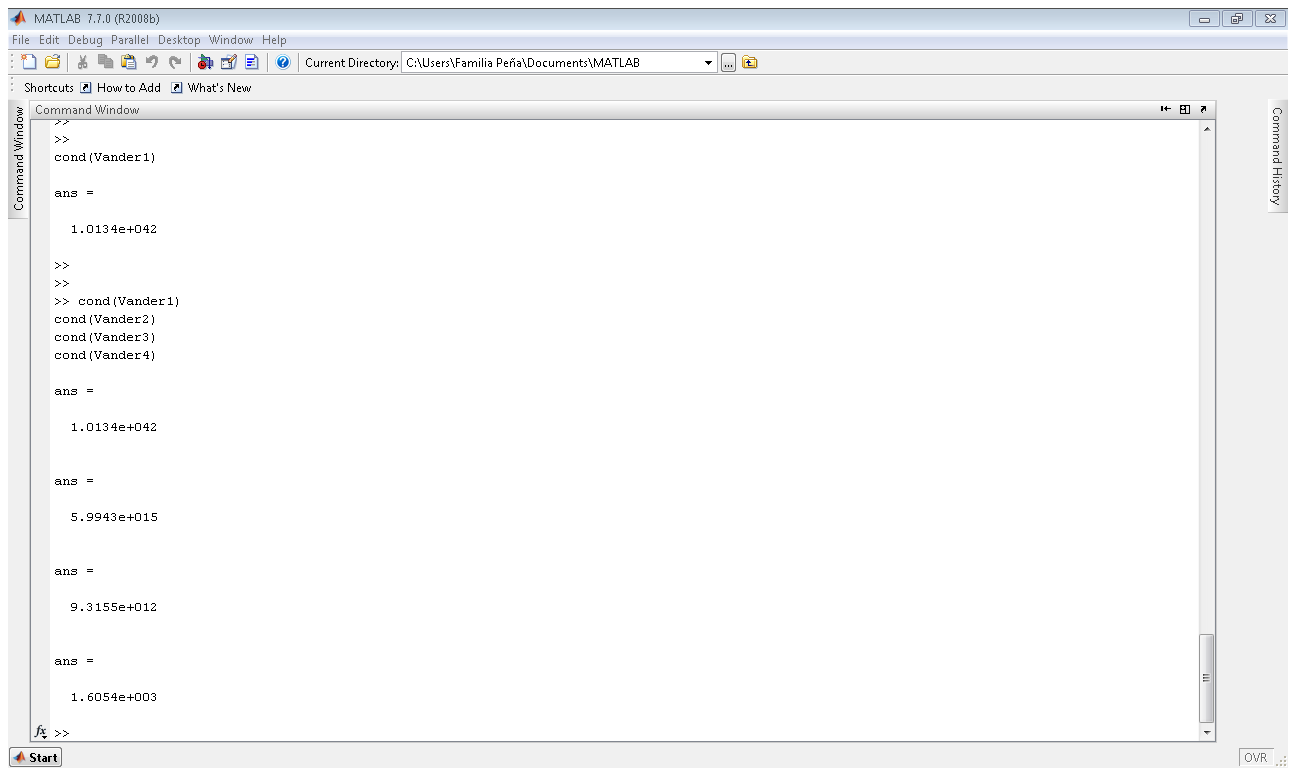


a) For each of these four sets of basic functions, generate the corresponding Vandermonde matrix and compute its condition number using a library routine for condition estimation. How do the condition numbers compare? Explain your results.



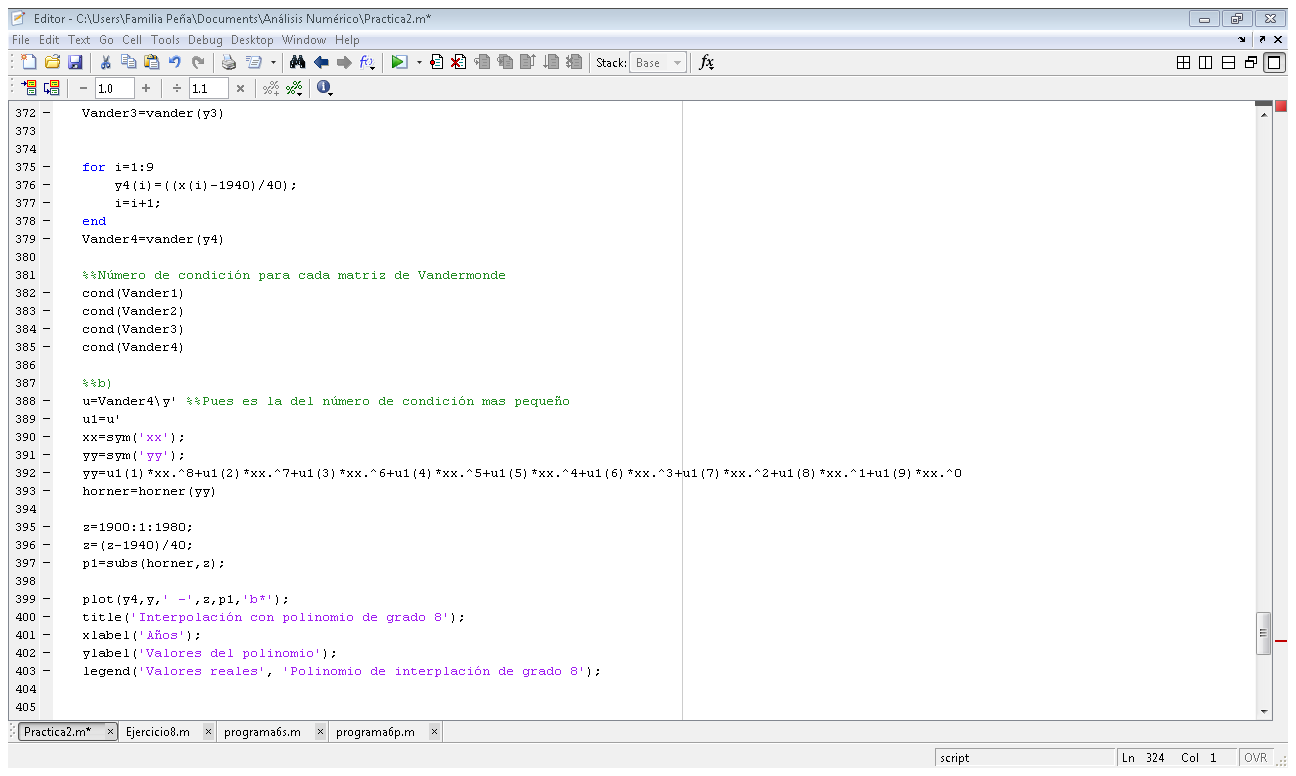


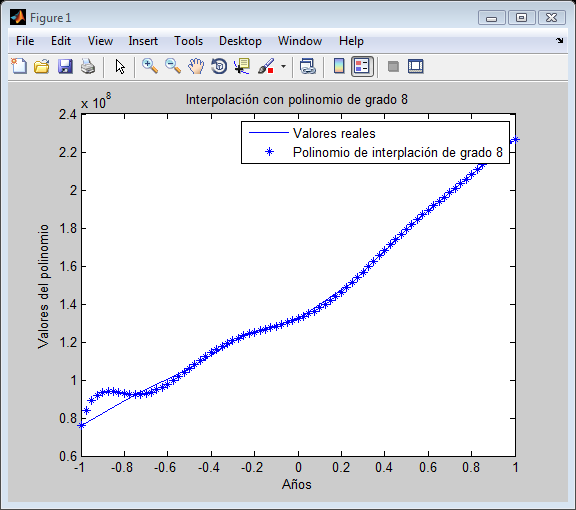




Se puede observar que el valor de las condiciones obtenidas disminuye de acuerdo con las funciones bases planteadas. Así, se puede concluir que la primera matriz es la peor condicionada y la mejor condicionada es la cuarta.

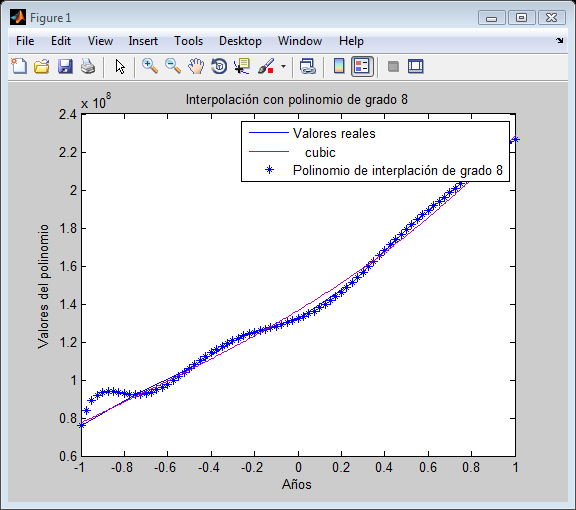
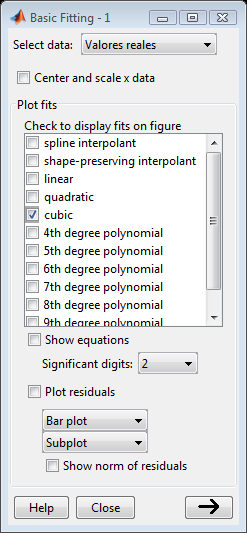
(b) Using the best-conditioned basis found in part a, compute the polynomial interpolant to the population data. Plot the resulting polynomial, using Horner’s nested evaluation scheme to evaluate the polynomial at one-year intervals to obtain a smooth curve. Also plot the original data points on the same graph.





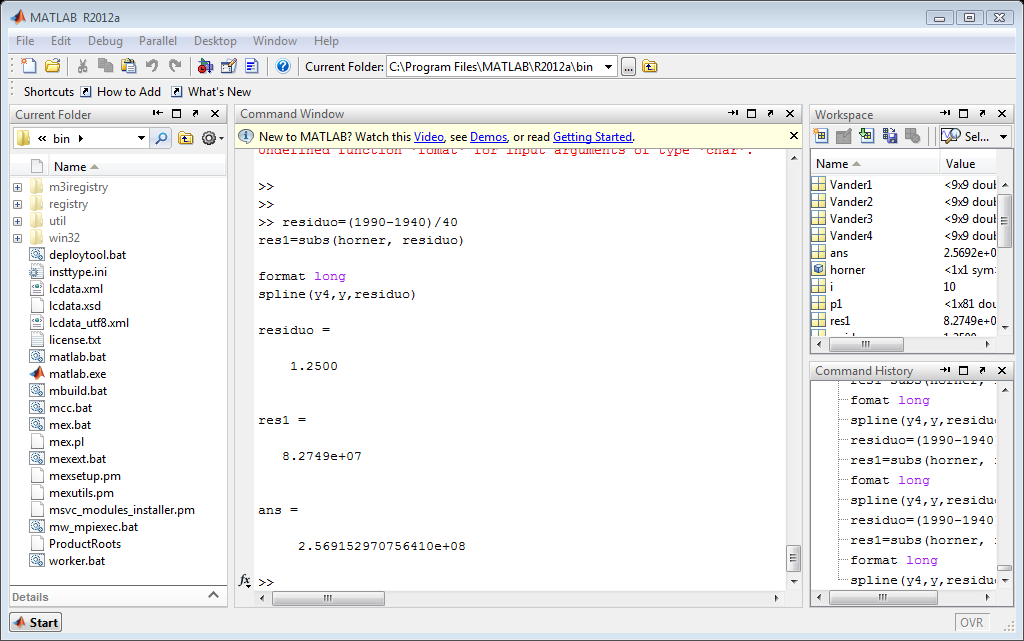
Por medio de la anterior gráfica, notamos que la transformación (t-1940)/40 se ajusta bien a los datos poblacionales, difiere levemente alrededor del (-1, -0.6) del intervalo.

(c) Use a cubic spline routine to interpolate the population data, and again plot the resulting curve on the same graph.



En esta gráfica notamos que la función cúbica se aproxima bastante bien a los valores reales, tanto que hasta pareciera que es una misma. Entonces el spline cúbico aproxima mejor que el polinomio de grado 8.

(d) Use both the polynomial and the spline to extrapolate the population to 1990 and compare the values obtained. How close are these to the true value of 248,709,873 according to the 1990 census?



Como estamos extrapolando, se hace un pronóstico. El pronóstico para cada uno al año 1990 es:

* Polinomio de grado 8: 82,749,141
* Spline cúbico: 256,921,529.7
* Censo: 204, 709,873

Si calculamos el error absoluto tenemos:

* Polinomio: 121,960,732
* Spline: 52,211,656.7

Ahora el error relativo:

* Polinomio: 0.5957
* Spline: 0.25505

Notamos que el error con respecto al valor exacto es menor con el spline cúbico, por lo que aseguramos que para la población de 1990 es mejor extrapolar con un spline cúbico.