OBLIGATORIO

1

Métodos Numéricos

Curso 2010

Grupo Nro.50

Germán Ruiz 4.317743-2

Martín Zanetti 4.585215-3

Federico Mujica 4.786543-9

INDICE

OBJETIVOS DEL OBLIGATORIO

Objetivo Parte 1

La primera parte consiste en la implementación de distintos algoritmos para la resolución del sistema lineal $Ax=b$ siendo A una matriz banda. Estudiar los tiempos de ejecución de los distintos algoritmos utilizando las propiedades de la matriz banda, para compararlos en relación al método por defecto utilizado por MatLab.

Objetivo Parte 2

Discretizar la ecuación diferencial dada, para luego expresarla como un sistema lineal. Aplicar los algoritmos implementados en la parte anterior para resolver este nuevo sistema en los distintos casos planteados. Resolver analíticamente las ecuaciones diferenciales y luego compararlas con las soluciones halladas, mediante los algoritmos, del sistema lineal que queda expresado a partir de las ecuaciones.

PARTE 1

Parte 1a

El algoritmo de eliminación gaussiana consiste en descomponer la matriz banda A en dos marices banda L y U, donde la matriz L es una triangular inferior y la matriz U es una triangular superior. Para poder implementar este algoritmo tomamos el ancho de banda horizontal de la matriz como la cantidad de diagonales que hay en la matriz contando después de la diagonal principal. Para el ancho de banda vertical realizamos lo mismo pero contando las diagonales hacia abajo. Implementamos el algoritmo de manera tal que solo recorra las diagonales que están en la banda de la matriz, omitiendo el resto de las diagonales ya que sus elementos son todos iguales a cero[1]. Asumimos que la matriz A es una matriz diagonal dominante por filas para de esta manera evitar la estrategia de pivoteo parcial.

Parte 1b

Partimos de un sistema lineal Ax=b. Luego de realizar la eliminación gaussiana de la parte a, tenemos la descomposición de la matriz A en las matrices L y U. Con esto cambiamos el sistema a resolver por LUx=b. Para resolver esto lo dividimos en 2 sistemas. El primero es el sistema Ly=b y el segundo es Ux=y. [2]Para resolver el primer sistema realizamos el método de sustitución hacia adelante aprovechando la estructura de L. Luego, con la solución y resolvemos el segundo sistema aplicando el método de sustitución hacia atrás, aprovechando la estructura de U. Devolvemos la solución x que es la solución del sistema original Ax=b.

[1] Ver el script EgMb para ver la implementación más en detalle
[2] Ver el script SustMb para ver la implementación del algoritmo.

Parte 1c

La estrategia del método de Thomas es un método diseñado para matrices tridiagonales. Consiste en la resolución del sistema Ax=b pero de manera tal que no se utiliza la matriz completa A sino solamente las tres diagonales principales, en forma de vectores. Realiza primero eliminación gaussiana y luego realiza las sustituciones correspondientes. Implementamos el algoritmo de tal forma que pueda resolver el sistema de la forma antes dicha.[3]

Parte 1d

En la implementación del método de Thomas se realizan operaciones de división y resta donde ocurren errores de punto flotante, por la tanto si un número implicado en estas operaciones se acerca a cero el método no es preciso. Por lo tanto si una matriz tiene determinante cerca de cero el método de Thomas no es aplicable.

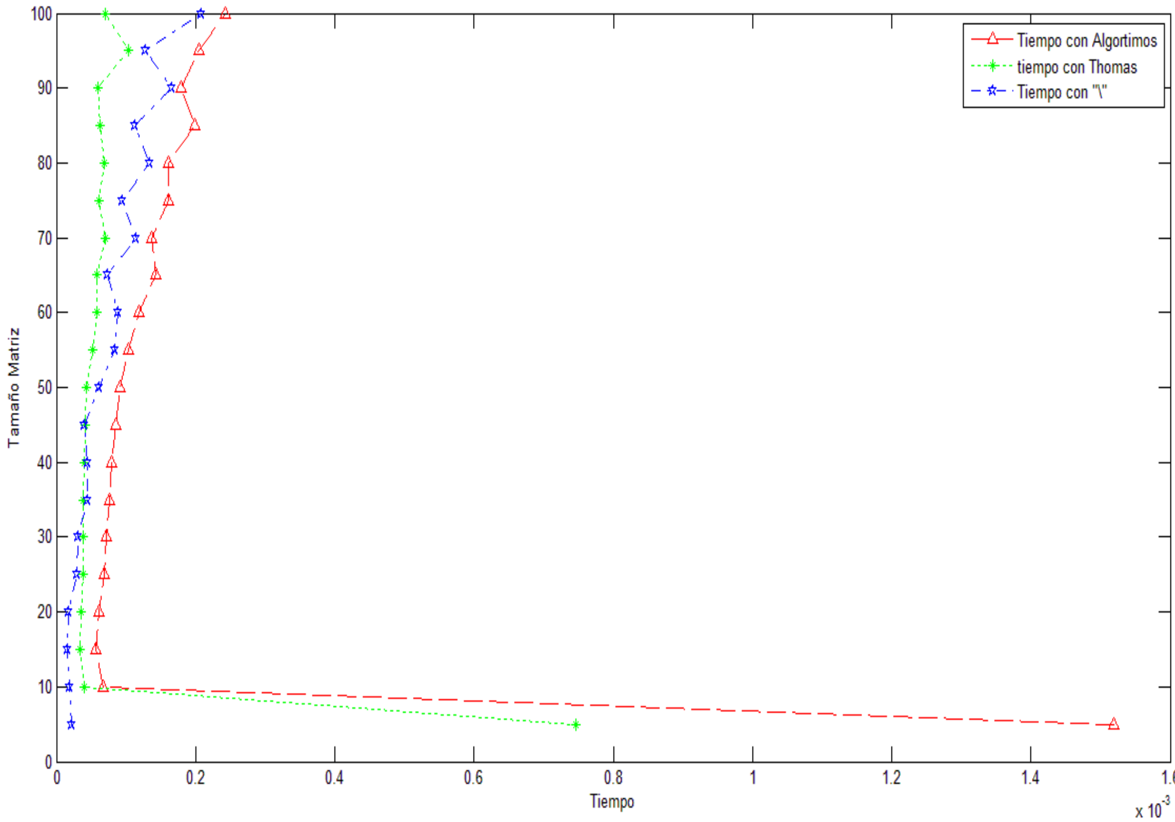
Parte 1e

Comparamos los tiempos de ejecución de los algoritmos con el comando "\" de MatLab para resolver el sistema Ax=b con distintas dimensiones de matrices banda. Con matrices con dimensión de 10^k, con k =4 es necesaria gran cantidad de memoria en la computadora por lo que no siempre se puede realizar y al ser solo cuatro puntos en la gráfica no se puede visualizar muy bien la diferencia entre uno y otro método. [4]Para apreciar mejor las diferencias realizamos comparaciones con matrices de dimensión 100, 500, 1000 y 5000.

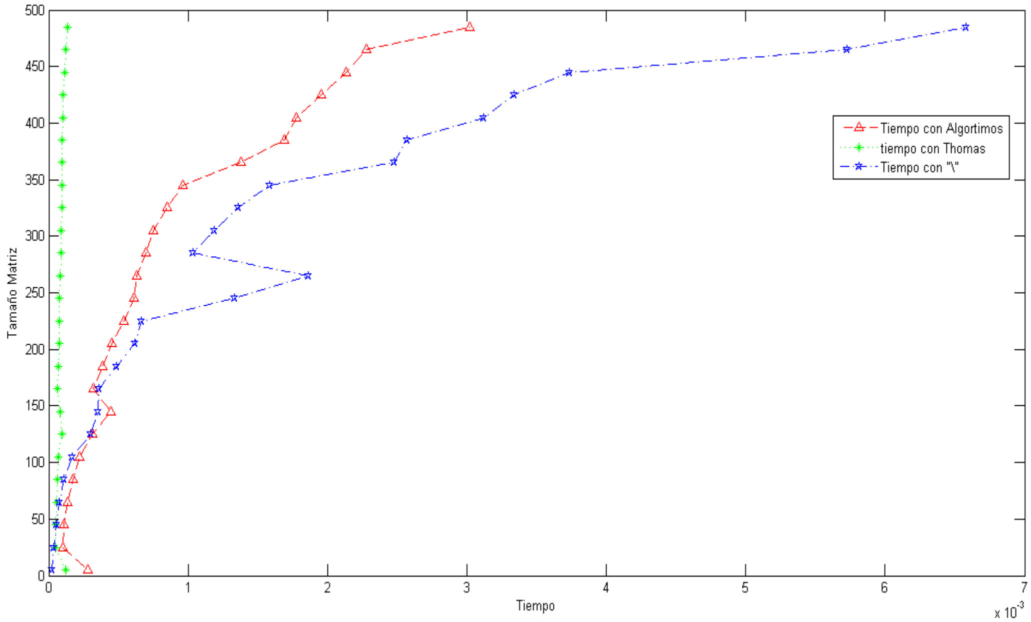
[3] Ver en los anexos el script Thomas para ver su implementación.

[4] Ver en los anexos las graficas realizadas y los scripts PruebaAlgoritmos y PruebaTiempo.

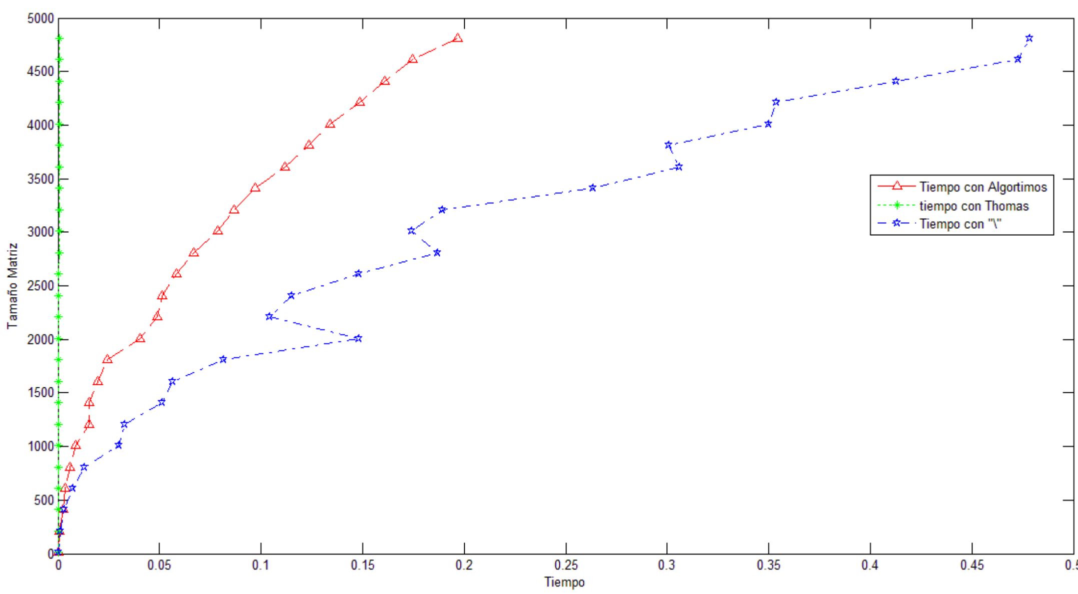
Grafica1:


Tiempo con Algortimos
tiempo con Thomas
Tiempo con "\"
Tamaño Matriz
Tiempo
x 10^-3
0
10
20
30
40
50
60
70
80
90
100
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6

Grafica2:



Grafica 3:



Conclusiones

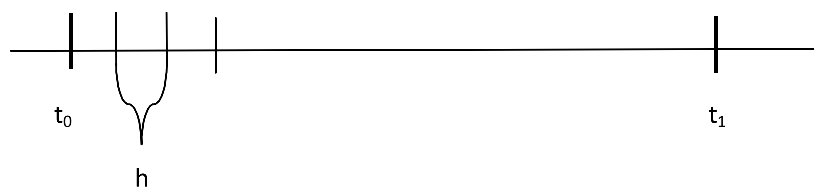
Observando la gráfica 1 podemos ver que para matrices de dimensión pequeña, menores que 30 el método por defecto de MatLab es más rápido que los restantes. El método de Thomas es mejor que el de los algoritmos de la parte a y b para éstas matrices. A partir de dimensión 30 el método de Thomas comienza a igualar al de MatLab mientras que el de los algoritmos sigue siendo peor que los dos anteriores. En dimensión 40 ya tenemos que el método de Thomas pasa a ser más rápido que el de MatLab, situación que mantiene hasta el final de la gráfica (dimensión 100). Estos dos métodos siguen siendo mejores que el de los algoritmos.

Observando la gráfica 2 vemos que para dimensiones entre 150 y 200, el método de los algoritmos de la parte a y b pasa a ser más rápido que el de MatLab, pero no supera el de Thomas, situación que se mantiene en dimensiones mayores (gráfica 3).

Podemos concluir que el método de Thomas es el más eficiente para resolver sistemas lineales con matrices grandes, con la desventaja de que solo es posible utilizar este método en matrices tridiagonales. Para las situaciones en que nos enfrentemos con matrices que no sean tridiagonales la mejor opción es utilizar el método de los algoritmos de la parte a y b cuando se traten de matrices superiores a 200, para otro caso es mejor utilizar el de MatLab. Para el caso que se traten con matrices pequeñas, de hasta dimensión 30 es mejor utilizar el método de MatLab, que aunque no exista mucha diferencia entre los tiempos de ejecución, éste método nos ahorra los problemas de tratar con matrices que no sean tridiagonales.

PARTE 2

Parte a



Definimos h = (t1-t0)/N con un N perteneciente a los enteros mayor que cero.

Definimos $X_i = t_0 + ih$, con i=0. . N

Para simplificar la notación nos definimos $y_i = y(Xi)$, $a_i=a(Xi)$, $b_i=b(Xi)$, $c_i=c(Xi)$.

Parte b

Usando cocientes incrementales discretizamos y'' por diferencia centrada como:

$y''(x) = (y(x+h) - 2y(x) + y(x-h))/h^2$

Para obtener error de truncamiento de orden 2 hacemos desarrollo de Taylor hasta orden 3:

$y(x+h) = y(x) + y'(x)h + y''(x)h^2/2! + y'''(x)h^3/3! + O(h^4)$

$y(x-h) = y(x) – y'(x)h + y''(x)h^2/2! – y'''(x)h^3/3! + O(h^4)$

Luego sumando las dos ecuaciones:

$y(x+h) + y(x-h) = 2y(x) + 2y''(x) h^2/2! + O(h^4)$

Despejando y'' y dividiendo entre h^2 obtenemos el orden 2 de error:

$y''(x) = (y(x+h) – 2y(x) + y(x-h))/h^2 + O(h^2)$

Usando las notaciones queda:

$y''(Xi) = (y_{i-1} – 2y_i + y_{i+1}) / h^2$

Parte c

Para obtener error de truncamiento de orden 2 aproximamos y' usando cocientes incrementales por diferencia centrada.

$y'(x) = (y(x+h) - y(x-h)) / 2h$

Hacemos Taylor hasta orden 2:

$y(x+h) = y(x) + y'(x)h + y''(x)h^2/2! + O(h^3)$

$y(x-h) = y(x) - y'(x)h + y''(x)h^2/2! + O(h^3)$

Luego restando las dos ecuaciones:

$y(x+h) - y(x-h) = 2y'(x)h + O(h^3)$

Dividimos entre h y despejamos y' obteniendo el error de truncamiento de orden 2:

$y'(x) = (y(x+h) - y(x-h))/2h + O(h^2)$

Luego usando las notaciones queda como:

$y'(X_i) = (y_{i+1} - y_{i-1})/2h$

Parte d

La ecuación diferencial es la siguiente:

$y''(X_i) + a(X_i)y'(X_i) + b(X_i)y(X_i) + c(X_i) = 0$, $y(X_0)=\alpha$, $y(X_N)=\beta$ con i=0..N

Sustituyendo los valores por los obtenidos en la parte b:

$(y_{i-1} - 2y_i + y_{i+1}) / h^2 + a_i (y_{i+1} - y_{i-1})/2h + b_i y_i + c_i = 0$

Multiplicando todo por ($2h^2$) y realizando cálculos se llega a:

$y_{i-1}(ha_i - 2) + y_i(4 - 2h^2b_i) + y_{i+1}(-2 - ha_i) = 2h^2c_i$ para i=1..N-1.

$$\begin{array}{l} X_0 = t_0 \\ X_N = t_1 \end{array} \quad \begin{cases} y_0 = y(X_0) = y(t_0) = \alpha \\ y_N = y(X_N) = y(t_1) = \beta \end{cases}$$

$$\begin{array}{ll}
i = 1 & y_1(4 - 2h^2b_1) + y_2(-2 - ha_1) = 2h^2c_1 - \alpha(ha_1 - 2) \\
i = 2 & y_1(ha_2 - 2) + y_2(4 - 2h^2b_2) + y_3(-2 - ha_2) = 2h^2c_2 \\
i = 3 & y_2(ha_3 - 2) + y_3(4 - 2h^2b_3) + y_4(-2 - ha_3) = 2h^2c_3 \\
\vdots & \vdots \\
i = N - 1 & y_{N-2}(ha_{N-1} - 2) + y_{N-1}(4 - 2h^2b_{N-1}) = 2h^2c_{N-1} + \beta(2 + ha_{N-1})
\end{array}$$

Luego expresándolo como un sistema lineal en forma matricial:

$$\begin{pmatrix}
4 - 2h^2b_1 & -2 - ha_1 & & & 0 \\
ha_2 - 2 & 4 - 2h^2b_2 & -2 - ha_2 & & \\
& ha_3 - 2 & 4 - 2h^2b_3 & -2 - ha_3 & \\
& & \ddots & \ddots & -2 - ha_{N-2} \\
0 & & & ha_{N-1} - 2 & 4 - 2h^2b_{N-1}
\end{pmatrix}
\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N-1} \end{pmatrix}
=
\begin{pmatrix} 2h^2c_1 - \alpha(ha_1 - 2) \\ 2h^2c_2 \\ 2h^2c_3 \\ \vdots \\ 2h^2c_{N-1} + \beta(2 + ha_{N-1}) \end{pmatrix}$$

Parte f

Resolución analítica de las ecuaciones diferenciales

Ecuación 1

$a(t) = b(t)=1 \quad c(t) = \sin(t) \quad t_0=0 \quad t_1=2\pi \quad \beta=\alpha=0;$

$$\Longrightarrow \begin{cases} y'' + y' + y + \sin(t) = 0 \\ y(0) = 0,\ y(2\pi) = 0,\ t \in [0, 2\pi] \end{cases}$$

Sol Gral (ELH):

$y'' + y' + y = 0$

$\longrightarrow \lambda^2 + \lambda + 1 = 0$

sol = $-(1/2) \pm (\sqrt{3})i/2$

$\longrightarrow h(t) = (C_1 e^{-(1/2)t} \cos((\sqrt{3}/2)t) + C_2 e^{-(1/2)t} \sin((\sqrt{3}/2)t))$

Sol Part (ELNH):

$f(t) = -\sin(t)$

$Q(t) = C\,\text{sen}(t) + D\cos(t)$

$Q'(t) = C\cos(t) - D\,\text{sen}(t)$

$Q''(t) = -C\,\text{sen}(t) - D\cos(t)$

Llegamos a:

$\longrightarrow \text{Sen}(t)(-D) + \cos(t)(C) = \text{sen}(t)(-1)$

$\longrightarrow D = 1; C = 0;$

solPart = cos(t);

Sol Gral:

$y = (C_1 e^{-(1/2)t} \cos((\sqrt{3}/2)t) + C_2 e^{-(1/2)t} \sin((\sqrt{3}/2)t)) + \cos(t);$

$$\begin{matrix} y(0) = 0 \\ y(2\pi) = 0 \end{matrix} \Rightarrow \begin{cases} C_1 + 1 = 0 \longrightarrow \quad C_1 = -1 \\ C_1 e^{-\pi} \cos(\sqrt{3}\pi) + C_2 e^{-\pi} \text{sen}(\sqrt{3}\pi) + 1 = 0 \end{cases}$$

$$C_2 = \frac{-1 + e^{-\pi} \cos(\sqrt{3}\pi)}{e^{-\pi} \text{sen}(\sqrt{3}\pi)}$$

Ecuación 2

$a(t) = 0$ $b(t)=1$ $c(t) = e^t$ $t_0=0$ $t_1=1$ $\beta=1$ $\alpha=0;$

$$\Rightarrow \begin{cases} y'' + y = -e^t \\ y(0) = 0,\ y(1) = 1 \quad t \in [0, 1] \end{cases}$$

Sol Gral (ELH):

$\lambda^2 + 1 = 0 \longrightarrow \lambda = \pm i$

$h(t) = (C_1 \cos(t) + C_2 \sin(t))$

Sol Part (ELNH):

$f(t) = -e^t$

$Q(t) = Be^t$

$Q'(t) = Be^t$

$Q''(t) = Be^t$

$Be^t + Be^t = -e^t \longrightarrow B = -(1/2)$

$solPart = -e^t/2$

Sol Gral:

$h(t) = (C_1 \cos(t) + C_2 \sin(t)) - e^t/2$

$$\begin{matrix} y(0) = 0 \\ y(1) = 1 \end{matrix} \Rightarrow \begin{cases} C_1 - 1/2 = 0 \longrightarrow C_1 = ½ \\ (½)\cos(1) + C_2 \operatorname{sen}(1) - e/2 = 1 \end{cases}$$

$$C_2 = \frac{2 - \cos(1) + e}{2\operatorname{sen}(1)}$$

$$\Rightarrow y(t) = \cos(t)/2 + \left[\frac{2 - \cos(1) + e}{2\operatorname{sen}(1)}\right] \operatorname{sen}(t) - (e^t/2)$$

Ecuación 3

$a(t)=t$ $b(t)=0$ $c(t)=0$ $t_0=0$ $t_1=1$ $\alpha=0$ $\beta=1$

$$\Rightarrow \begin{cases} y'' + ty' = 0 \\ y(0)=0 \quad y(1)=1 \quad t \in [0, 1] \end{cases}$$

(ELH con coeficientes no constantes)

C.V. $x = y' \longrightarrow x' + tx = 0$

$x' = -tx \longrightarrow x'/x = -t$

$\int 1/x \, dx = -\int t \, dt \longrightarrow Ln(x) = -t^2/2 + C$

$\longrightarrow x = e^{-t^2/2 + C}$

Deshacemos C.V

$y = \int e^{-u^2/2 + C} du$ entre t_0 y t

$\Rightarrow y(t) = \int e^{-u^2/2 + C} du$ entre 0 y t

Haciendo otro cambio de variable $u = z\sqrt{2} \longrightarrow du = \sqrt{2}dz$

Cambiando los limites de integración queda:

$y(t) = \int e^{-z^2 + C}\sqrt{2}dz$ entre 0 y $t/\sqrt{2}$

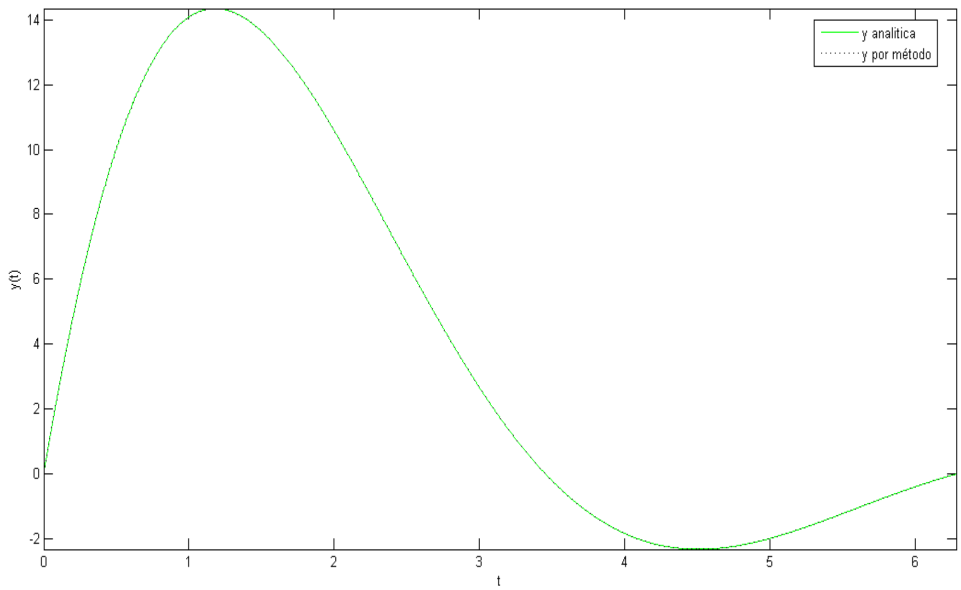
$\Rightarrow y(t) = e^{C}\sqrt{2}\int e^{-z^2}dz$ entre 0 y $t/\sqrt{2}$

[5]Usando la función erf() de Matlab y la condición inicial y(1)=1 despejamos C.

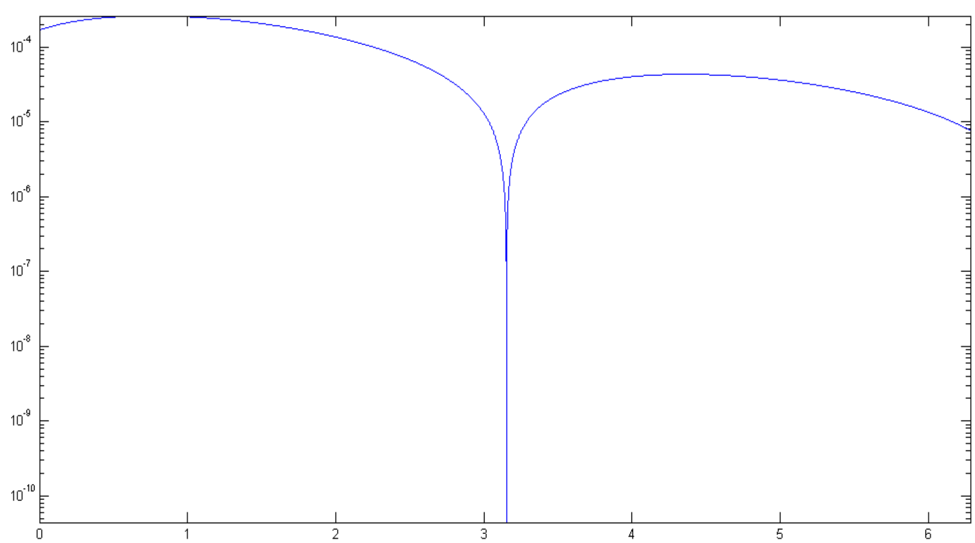
Con los algoritmos de la parte1 hallamos las soluciones del sistema lineal expresado y las comparamos con las soluciones halladas analíticamente.[6]

[5] Para ver los cálculos realizados ver en los scripts Parte2f.
[6] Ver los scripts funcion1, funcion2, funcion3,Comparacionesfunciones.

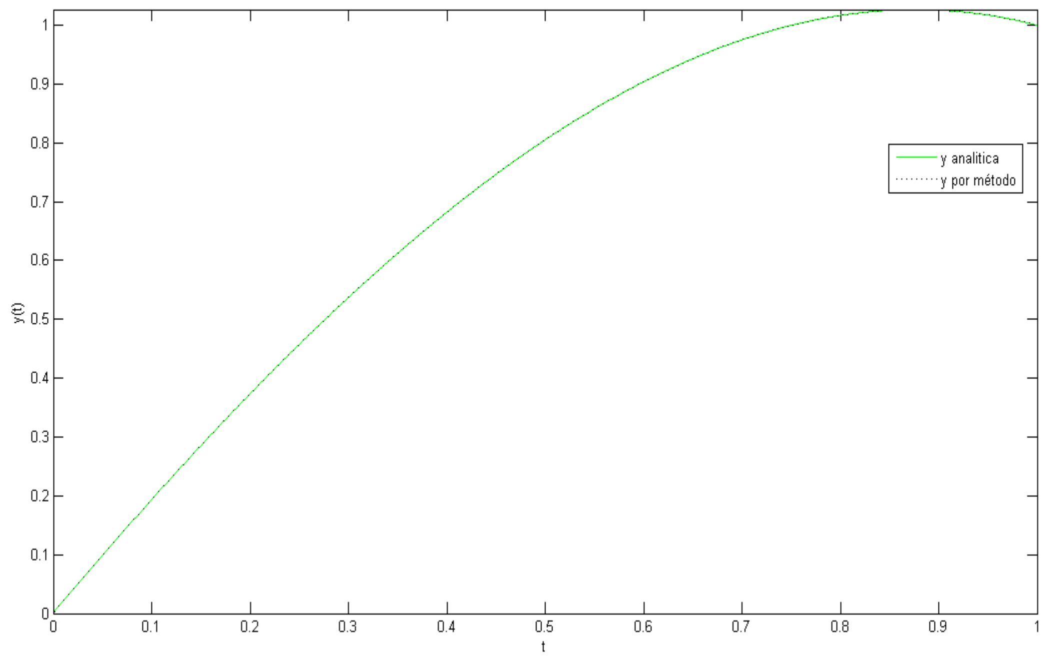
Grafica 1: Comparación entre la solución hallada en la parte e)1 y f)1.



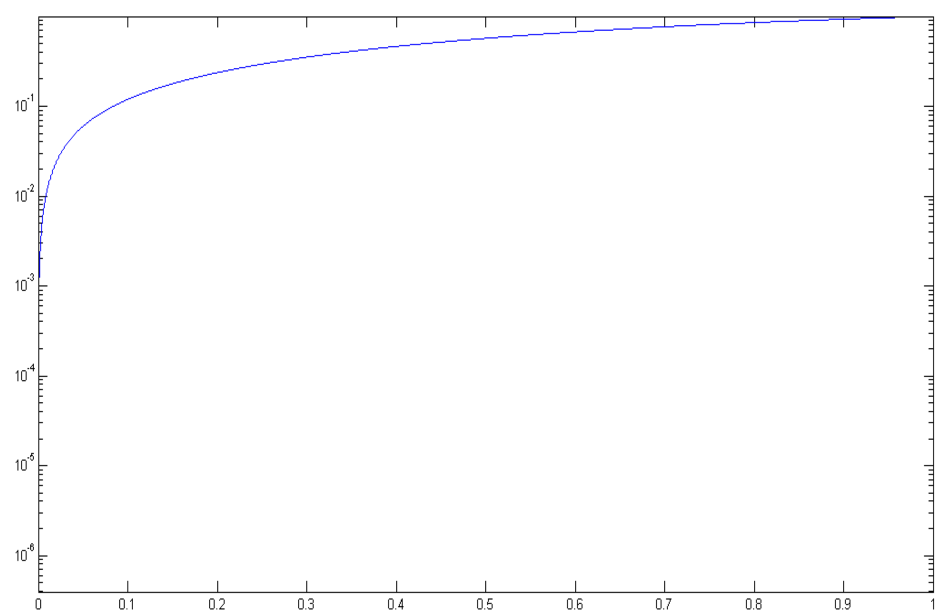
Error 1: Diferencia de las soluciones en escala logarítmica.



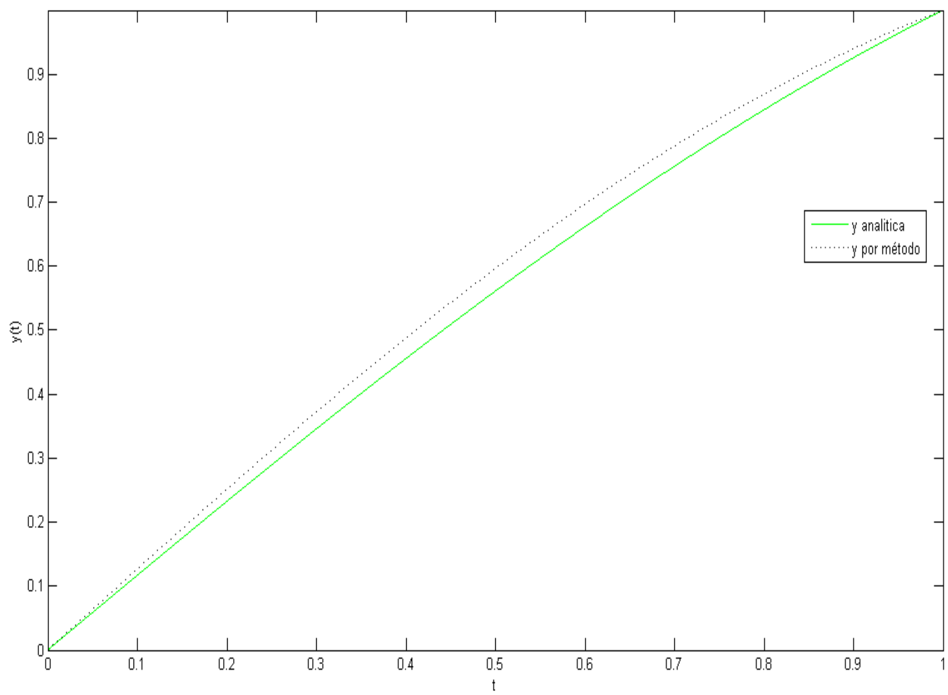
Grafica 2: Comparación entre la solución hallada en la parte e)2 y f)2.



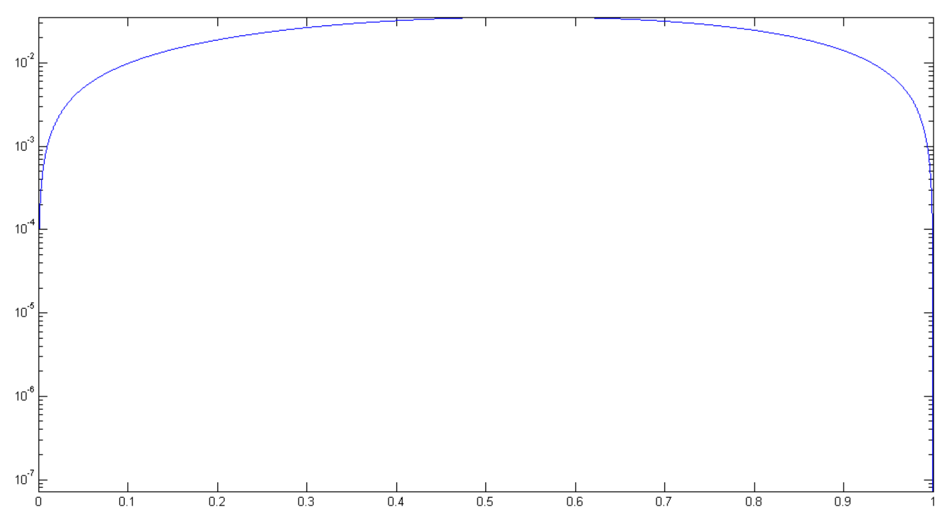
Error 2: Diferencia de las soluciones en escala logarítmica.



Grafica 3: Comparación entre la solución hallada en la parte e)2 y f)2.



Error 3: Diferencia de las soluciones en escala logarítmica.



Conclusiones

Como se puede observar en la gráfica 1 la solución de la ecuación diferencial mediante el método analítico es similar a la solución utilizando el método de Thomas para el sistema resultante de la parte d. En la gráfica se observa que prácticamente se superponen las curvas de la funciones soluciones pero esto no es del todo cierto como se puede corroborar mediante la gráfica de error 1. En ella se aprecian los errores del método utilizado, que son muy pequeños (con mínimo error en el orden de 10^{-11} y máximo en el orden de 10^{-4}).

En la gráfica 2 se observa un comportamiento similar al caso anterior con la diferencia que hay menor rango de error (con mínimo error en el orden de 10^{-7} y máximo en el orden de 10^{-5}). Esos errores se pueden apreciar en la gráfica de error 2.

En la gráfica 3 se puede observar una diferencia entre las curvas de las soluciones por lo que el error en este caso es mayor que en los casos anteriores. Esto se confirma con la gráfica de error 3 observando que el máximo error esta en el orden de 10^{-1} mientras que el mínimo esta en el orden de 10^{-8} haciendo que este caso sea el que tenga más rango de error. Este rango de error mayor debe ocurrir porque para resolverla analíticamente hay que usar la función erf() que probablemente cometa un error considerable.

Conclusiones generales

Luego de realizada la actividad y analizar los resultados concluimos que haciendo la combinación de la eficiencia del algoritmo de Thomas para resolver sistemas lineales con matrices tridiagonales de gran tamaño que concluimos en la parte 1 (tanto en rapidez como en ahorro de memoria), con el método realizado para expresar una ecuación diferencial dada como un sistema de la parte 2, nos garantiza una solución bastante precisa en comparación a la solución hallada analíticamente, y esto se ve en el escaso margen de error que fue detectado.

Por lo tanto concluimos que este método numérico es bastante preciso y eficiente para hallar soluciones de ecuaciones diferenciales ordinarias de orden 2.