



Centro de Investigación en Matemáticas, A.C.

Descripción tarea 12 - Métodos numéricos

Erick Salvador Alvarez Valencia

18 de Noviembre de 2017

1. Introducción

En el presente reporte se describirán los resultados obtenidos del método trapezoidal que se utilizó para resolver un problema de valor inicial (PVI) con una ecuación diferencial de segundo orden. Se describirá un poco la descripción del método utilizado para resolver el PVI así como ejemplos de los resultados obtenidos.

2. Descripción

El método consistió en realizar una discretización en un intervalo cerrado y convertir el PVI en un sistema de matricial, posteriormente se aplicó el método del trapecio en el mismo y se obtuvo un método implícito en donde se debió despejar el valor de y_{i+1} con lo que se llegó a un sistema de ecuaciones que es resoluble con algún método ya sea directo o iterativo, para el presente se utilizó el método de factorización LU.

A continuación se muestra el PVI:

$$y'' = -y + xy' - 2x \cos x$$

Con $y(0) = 0$ y $y'(0) = 3$.

La solución analítica de dicho problema es: $y(x) = x + 2 \sin x$.

El sistema de ecuaciones al cual se llegó con el método del trapecio se muestra en las hojas adjuntas al programa.

3. Ejemplo de ejecución

El programa se ejecutó con diferentes tamaños de la discretización, todos dentro del intervalo $[0, 5]$ y dentro de las ejecuciones se registró el error absoluto promedio y el error absoluto máximo, estos en comparación con el resultado de la solución analítica. De la misma forma se imprimió el último valor numérico calculado y el valor analítico en ese mismo x_i . A continuación se muestran dos de los resultados obtenidos con diferentes valores de n .

```
ericksav22@Erick-A-UB-GL552VM:~/Documentos/Maestria/GIT/Semestre 1/Métodos numéricos/Tarea 12/tarea12_ErickAV$ ./main 400
Error absoluto promedio: 0.0239453
Error máximo: 0.492171
Yn (numérico): 3.574323, Yn (analítico): 3.082151
ericksav22@Erick-A-UB-GL552VM:~/Documentos/Maestria/GIT/Semestre 1/Métodos numéricos/Tarea 12/tarea12_ErickAV$
```

(a) Figura 1. Resultados del programa, $n = 400$.

```
ericksav22@Erick-A-UB-GL552VM:~/Documentos/Maestria/GIT/Semestre 1/Métodos numéricos/Tarea 12/tarea12_ErickAV$ ./main 4000
Error absoluto promedio: 0.000233486
Error máximo: 0.00491211
Yn (numérico): 3.087064, Yn (analítico): 3.082151
ericksav22@Erick-A-UB-GL552VM:~/Documentos/Maestria/GIT/Semestre 1/Métodos numéricos/Tarea 12/tarea12_ErickAV$
```

(b) Figura 2. Resultados del programa, $n = 4000$.

Se puede ver en la Figura 2. que pese a que aumentó mucho el número de valores en la discretización, la solución tuvo una notable mejoría en comparación con la analítica, así mismo los errores se hicieron más pequeños.

4. Compilación y ejecución

Para compilar: En la carpeta encontraremos los archivos `.c` y `.h` con los que se podrá compilar el ejecutable. De la misma forma, en conjunto con los archivos anteriores, también podremos encontrar un Makefile para, en caso de encontrarse en linux, compilar de manera sencilla.

1. **Compilar usando Makefile:** En la terminal, nos colocamos en el directorio donde se encuentre el programa, y ejecutamos el comando `make`, automáticamente se realizará la compilación y se generará el ejecutable. El Makefile también contiene el comando `make help` el cual mostrará todas las opciones disponibles.
2. **Compilar directamente:** De la misma forma, podemos compilar directamente usando los siguientes comandos (en terminal):

- `gcc -c main.c -o obj/main.o`
- `gcc -c memo.c -o obj/memo.o`
- `gcc -c matriz_vector.c -o obj/matriz_vector.o`
- `gcc -c met_num.c -o obj/met_num.o`
- `gcc -o main obj/main.o obj/memo.o obj/matriz_vector.o obj/met_num.o -lm`

Para ejecutar: Únicamente debemos de usar el comando `./main` para ejecutar el programa en consola, este recibe el siguiente argumento:

- **Un número entero:** El tamaño de la discretización.

El programa ejecutará el método trapezoidal para el PVI mostrado anteriormente dentro del intervalo $[0, 5]$. Se mostrará como resultado el error absoluto promedio y el error absoluto máximo, así como el último valor y_n y el valor de la solución analítica.

Ejemplo de ejecución: `./main 400`