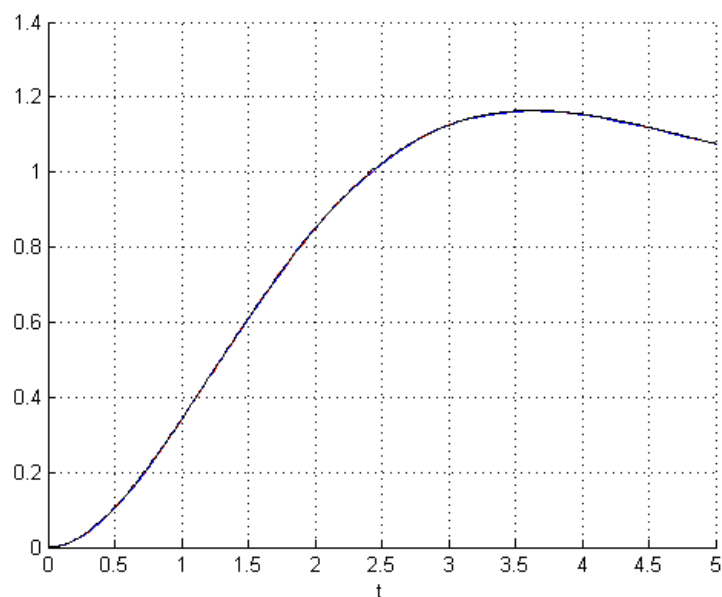


# Práctica 10

## Resolución de EDO con el Método de Diferencias Finitas



**Héctor Garbisu Arocha**

Curso 2015/16

Métodos Numéricos para la Computación

Grado en Ingeniería Informática

Escuela de Ingeniería Informática

Universidad de Las Palmas de Gran Canaria

# Índice

1. Ensamblado en C/C++ y tridiagonal en LAPACK ..... pág. 3
2. Comparativa en problemas con condiciones de inicio ..... pág. 4

# 1. Ensamblado en C/C++ y tridiagonal en LAPACK

El primer ejercicio plantea un problema de ecuaciones diferenciales que deberemos resolver mediante el método de diferencias finitas con un programa en C++, utilizando la librería ya conocida, LAPACK.

Para resolver el problema se ofrecen dos soluciones “y” (sin derivar) par dos valores de “x”, por lo que el tipo de contorno es 1(CONTOUR\_TYPE1). La forma en que esto afecta a la resolución del problema es que en el ensamblado de la matriz se ponen los valores de los extremos, en lugar de alguna combinación de valor y derivada.

Los vectores A, B, C y D son los coeficientes en cada punto (N puntos). Se pasan a la función solve como una sola matriz “params”.

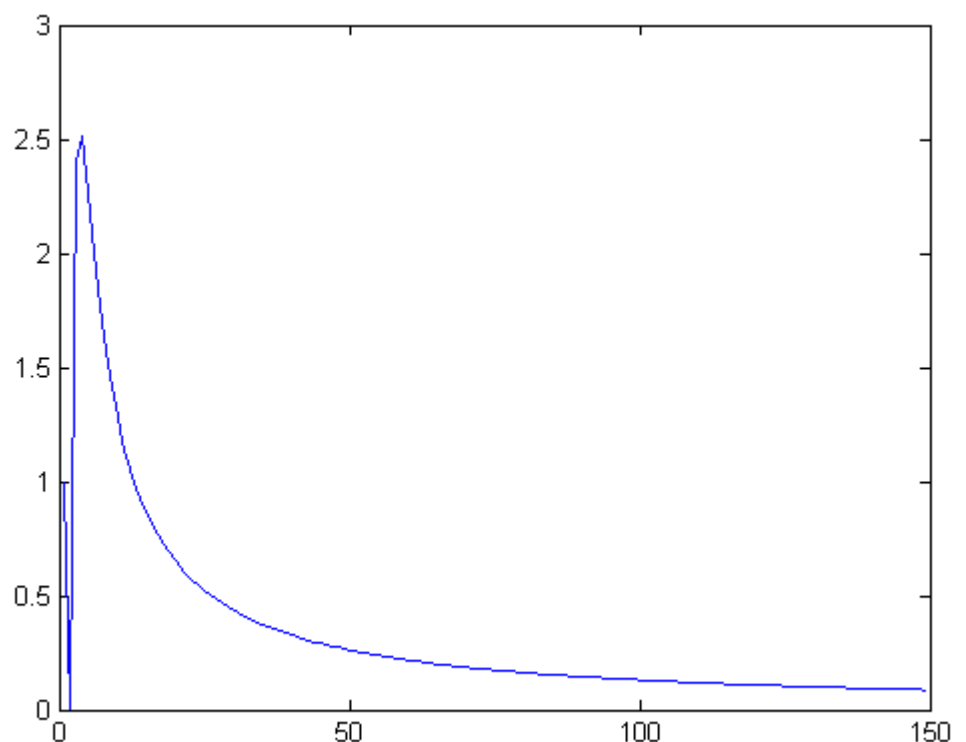
El error obtenido para diferentes valores de N va decreciendo a medida que aumenta el número de puntos de cálculo.

Para visualizar rápidamente el progreso hice un bucle para ir incrementando poco a poco N.

Estos son los 10 primeros valores y los dos últimos. Para un valor de 1 no tiene sentido hacer la aproximación por contornos (el resultado era infinito).

A la derecha, la curva para los primeros 150 valores

2	0
3	2.39877
4	2.51614
5	2.26957
6	2.00075
7	1.76791
8	1.57507
9	1.4161
10	1.28414
...	
148	0.0882158
149	0.087623



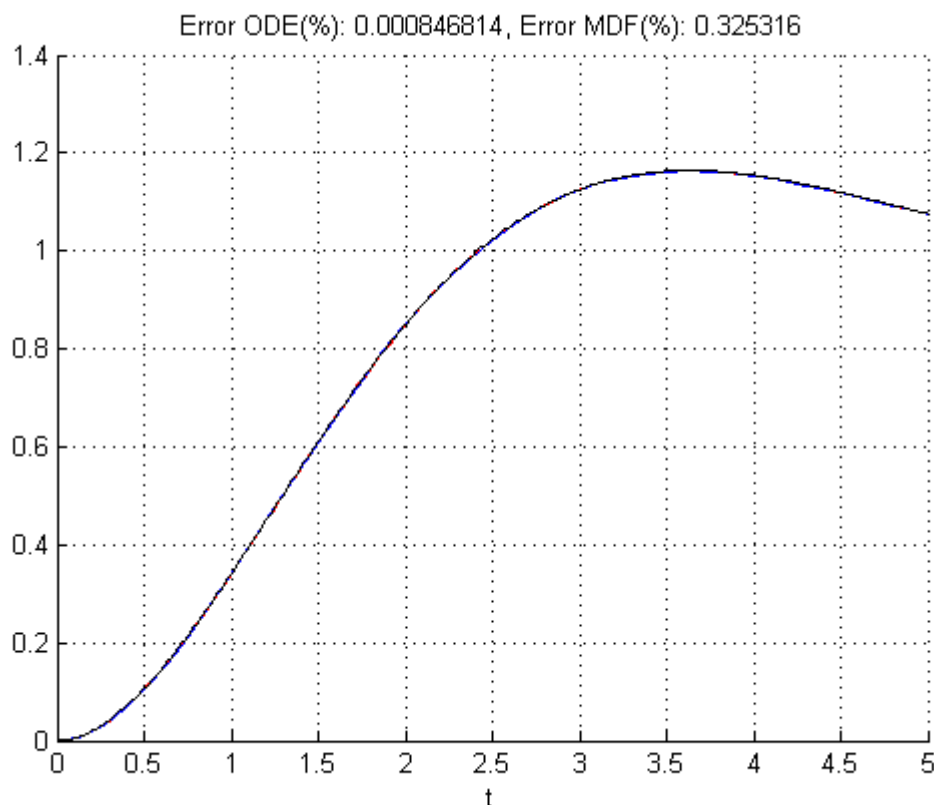
La aproximación es exacta para 2 valores porque una vez evaluados no hay que hacer ningún calculo aproximado de nada. Me sorprende que haya un aumento en el error entre 2 y 4 puntos.

## 2. Comparativa en problemas con condiciones de inicio

Ahora vamos a comparar cómo se comporta la resolución tipo Runge-Kutta (o Norman-Prince en ode45) con la resolución por método de diferencias finitas. Ya sabemos que no es un enfrentamiento nada justo usar condiciones de contorno con un método de condiciones de inicio, ni usar condiciones de inicio con un método de contorno.

Por eso, la aproximación a un escenario justo es simplemente forzar que los dos algoritmos terminen usando el mismo número de puntos intermedios.

Al ejecutar al programa, la gráfica no deja ver diferencias apreciables. El error medio calculado y que se muestra en el título, indica que el método ode45 ha sido mucho más preciso que el método de diferencias finitas. Cabe añadir que MDF no es bueno en esta clase de ejercicio.



Probablemente por eso, ode45 es la opción por defecto para EDOs en Matlab.