

Tarea 3

CE-3102 Análisis Numérico para Ingeniería

Estudiantes:

Juan Pablo Brenes Coto
Pablo Bustamante Mora
Emily Sancho Murillo

Profesor:

Prof. Dr. Pablo Alvarado Moya

28 de Agosto de 2018

1. Resultados del benchmark

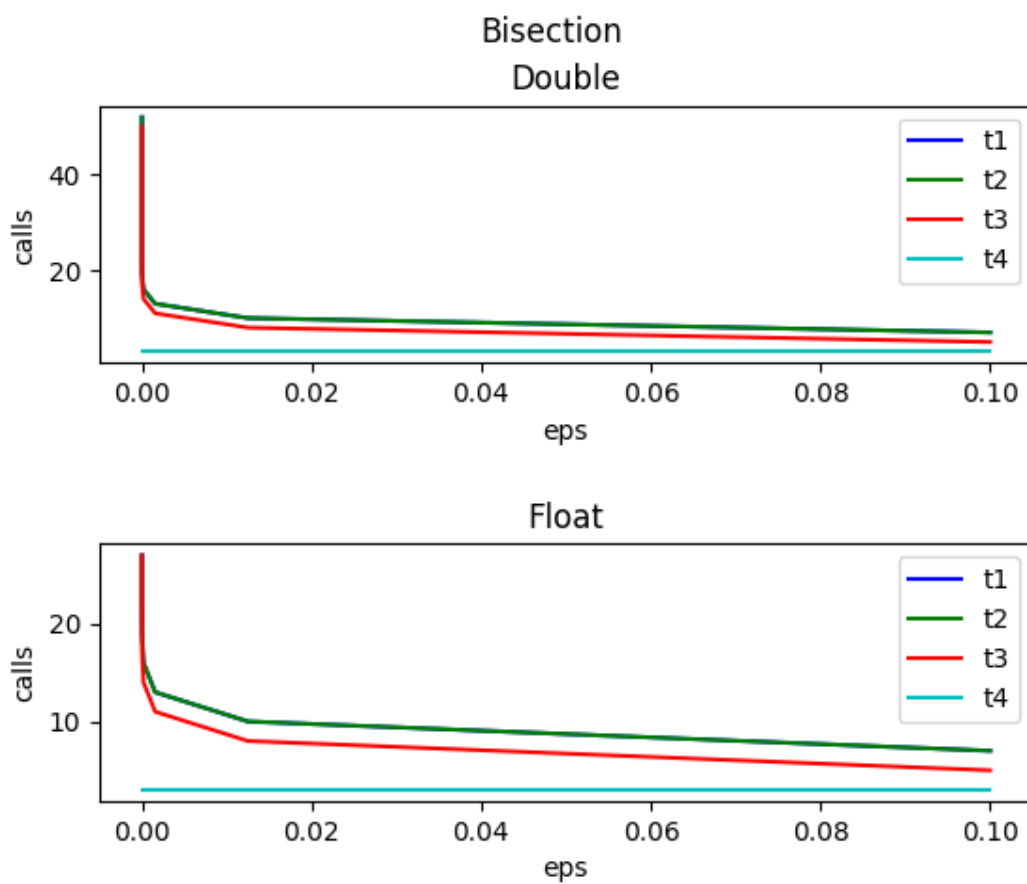


Figura 1: Número de llamadas en función de la tolerancia para el método de Bisección

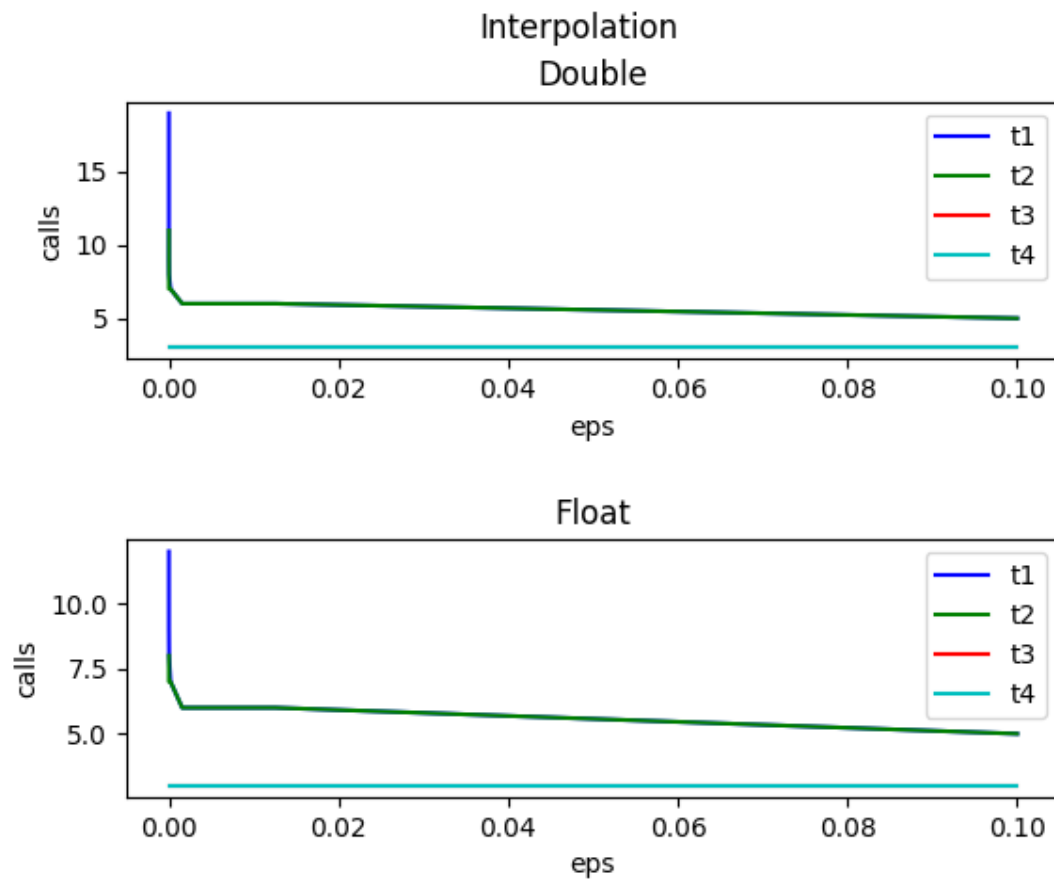


Figura 2: Número de llamadas en función de la tolerancia para el método de Interpolación

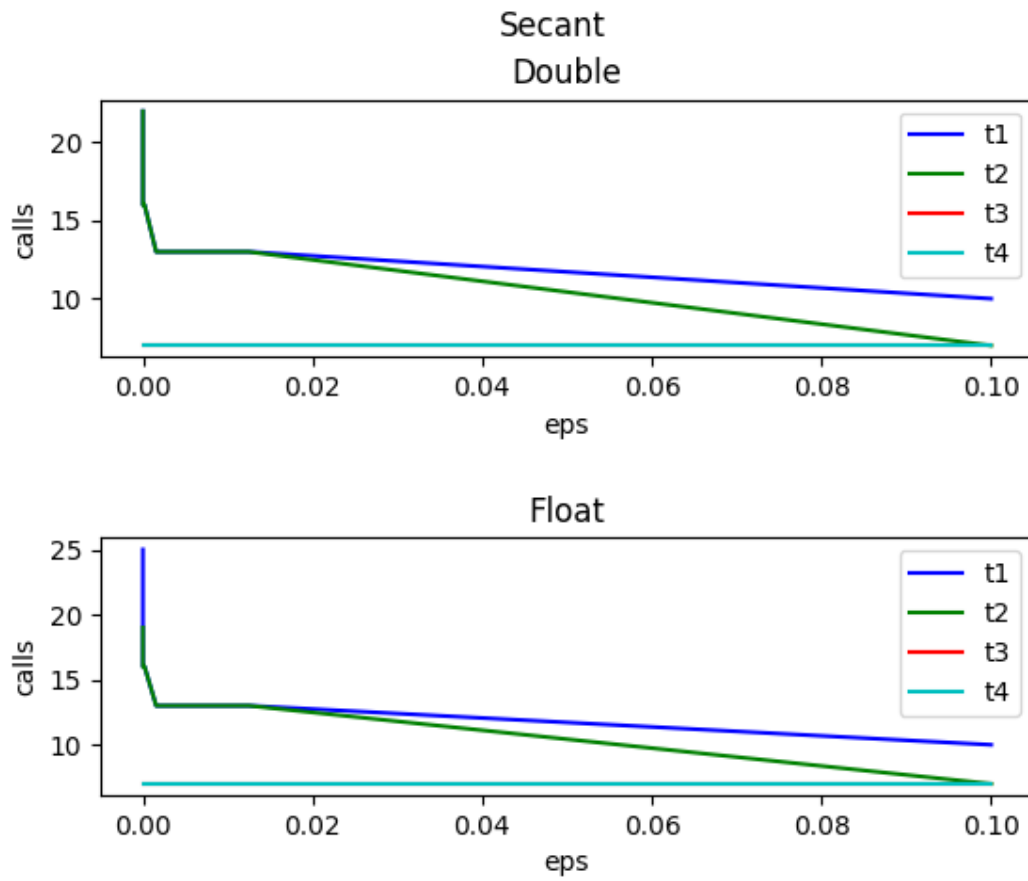


Figura 3: Número de llamadas en función de la tolerancia para el método de Secante

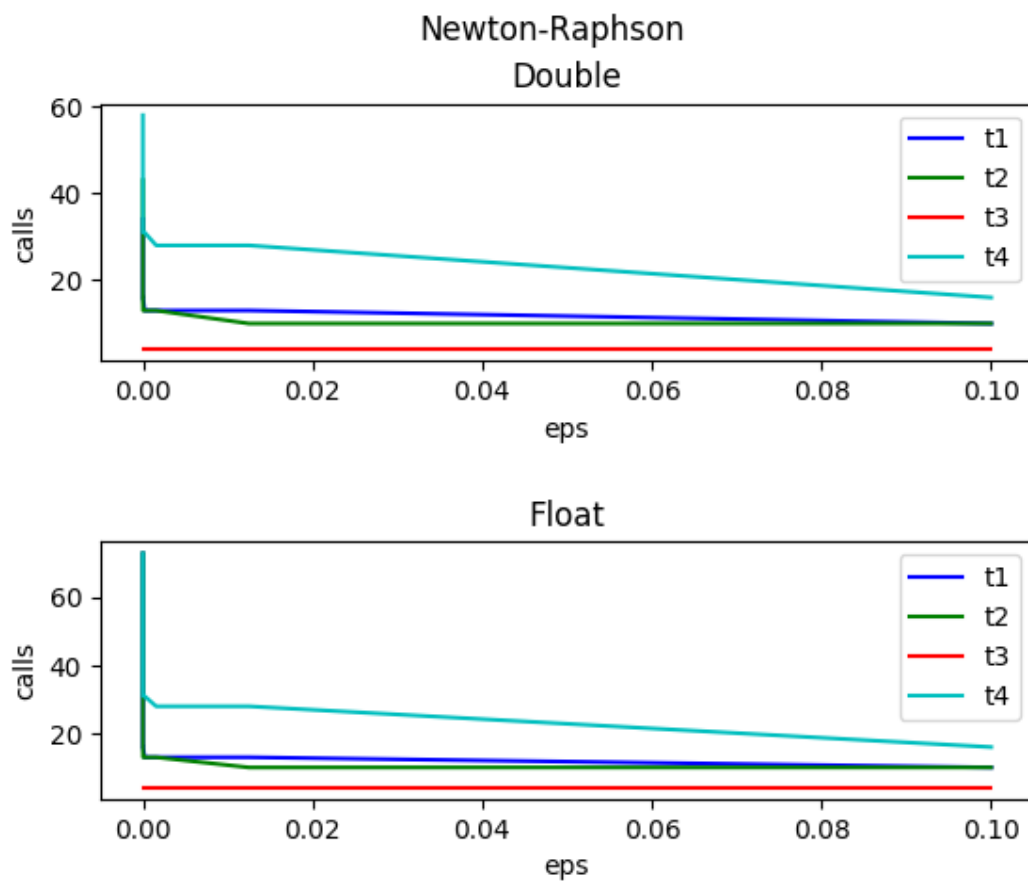


Figura 4: Número de llamadas en función de la tolerancia para el método de Newton-Raphson

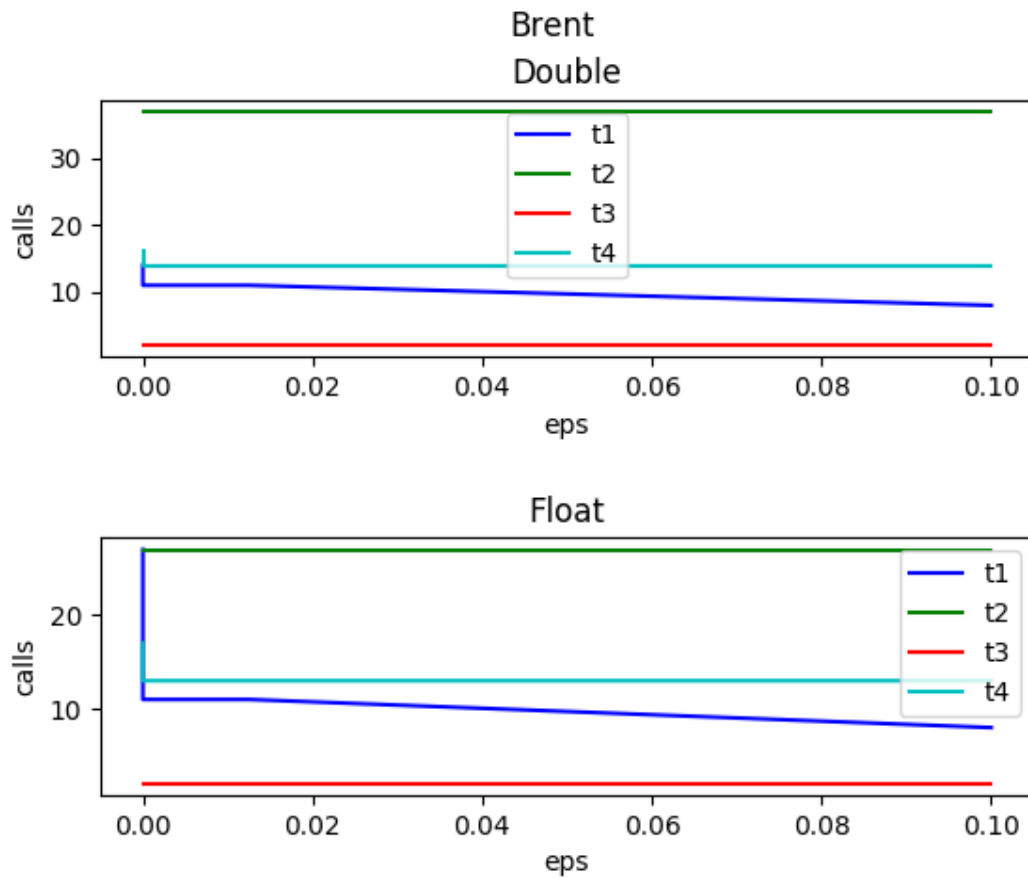


Figura 5: Número de llamadas en función de la tolerancia para el método de Brent

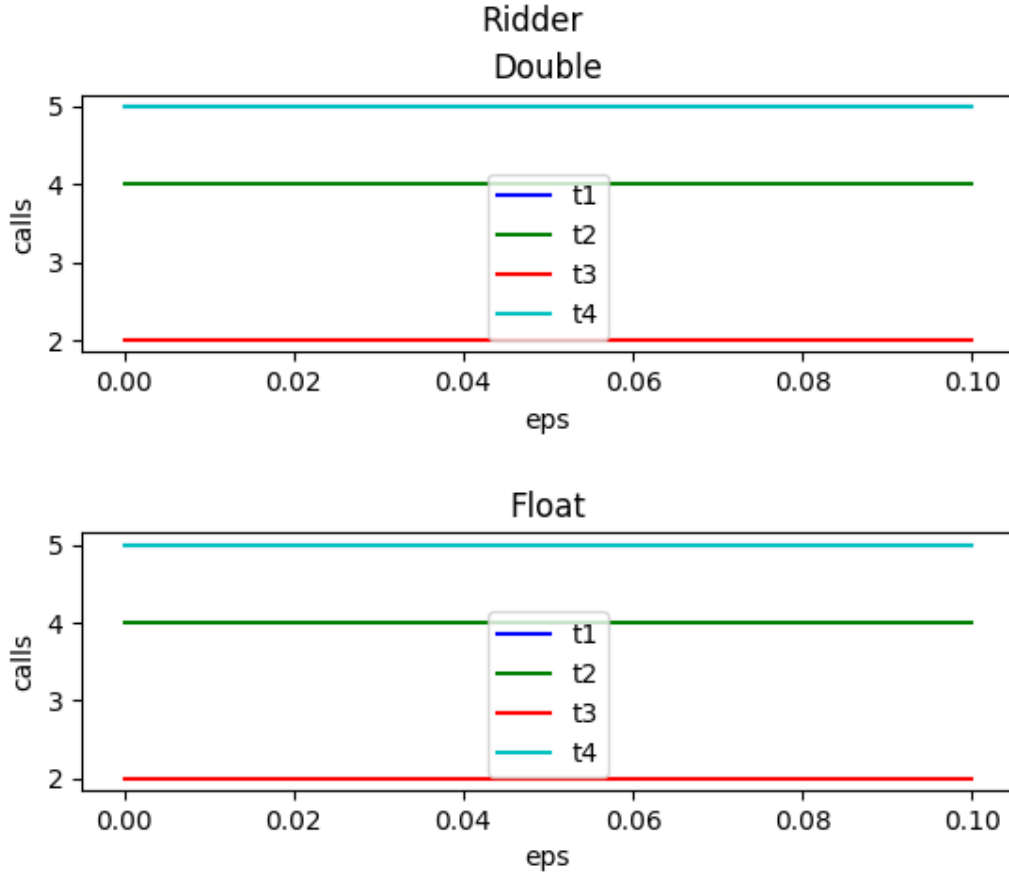


Figura 6: Número de llamadas en función de la tolerancia para el método de Ridder

2. Aumento del contador de llamadas a `operator()`

```
f_type c1(CallCounter<T>(t1<T>));
solver(c1, T(0), eps);
std::cout << c1.template target< CallCounter<T> >()->counter() << ";\n";
```

Debido a que "c1" es una instancia de "f_type", que es un objeto del tipo `std::function< T >` que almacena una instancia de la clase "CallCounter < T >", al ser esta ultima un template; es necesario utilizar la desambiguación ".template" para que el compilador identifique que se esta llamando al objeto almacenado dentro de la clase envolvente `std::function< T >` y no un método de esta ultima, lo cual implicaría un error.

Por otro lado, "target" si es una función miembro de `std::function< T >`. Esta retorna un puntero al objeto almacenado, que en este caso es la instancia de clase CallCounter; con este puntero es posible acceder a la función "counter()" de dicha clase, la cual permite obtener el número de llamadas a "operator()". Si no se utilizara ".template" el compilador podría interpretar que "counter()" es una función miembro de `std::function< T >`. En cuanto a la variable utilizada para llevar la cuenta de las llamadas a "operator()", se utiliza el modificador "mutable" para permitir que esta cambie de valor una vez que a sido inicializada en el constructor de la clase CallCounter.