

Exercise 21 - Arctangent and Arccotangent

Simon Vendelbo Bylling Jensen

March 17, 2019

Abstract

The "Big programming exercise" solution of exercise 21. An implementation of Arctangent and Arccotangent using complementary and mutual recursion methods for a more efficient algorithm by solving their generating differential equation.

1 Problem description

I have the student number 201507956, where 56 with modulus 35 (including the 0th exercise) will give me exercise 21. This problem is regarding solving the generating differential equations for Arctangent and Arccotangent. The two differential equations to implement, i

$$\arctan(x) = \int_0^x \frac{1}{z^2 + 1} dz, \quad (1)$$

$$\operatorname{arccot}(x) = \int_x^\infty \frac{1}{z^2 + 1} dz. \quad (2)$$

This should at first be introduced, and reduced to numerical solution for $x > 0$. Afterwards by complementary relation:

$$\arctan(x) + \operatorname{arccot}(x) = \frac{\pi}{2}. \quad (3)$$

This should be implemented for most efficient calculation, by solving $\arctan(x)$, if $x \leq 1$, and $\operatorname{arccot}(x)$ for $x > 1$. The algorithm should be done similarly to the error-function exercise by solution of the differential equation, which the integral equation represents. As an optional exercise, one can examine whether $x = 1$ is the optimal point to switch the integration routine.

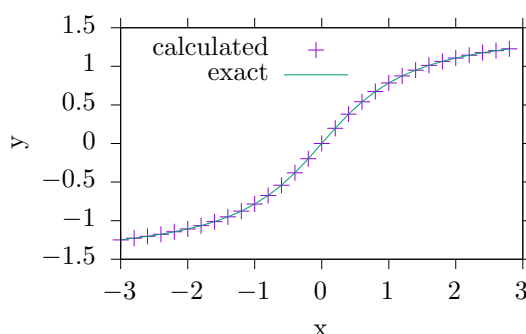


Figure 1: Comparison between the calculated arctangent function using the differential equation found in "myarctan.c" and the arctangent function from the math.h library.

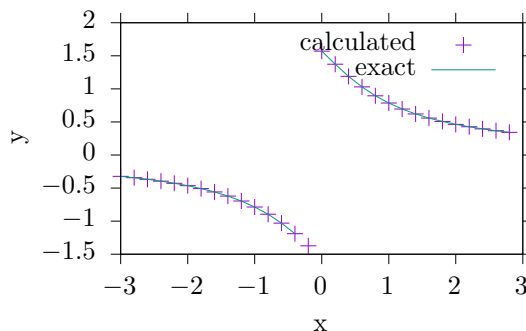


Figure 2: Comparison between the calculated arccotangent function using the differential equation found in "myarccot.c" and the arccot function from the math.h library, by calculation of arctan of the inverse value.

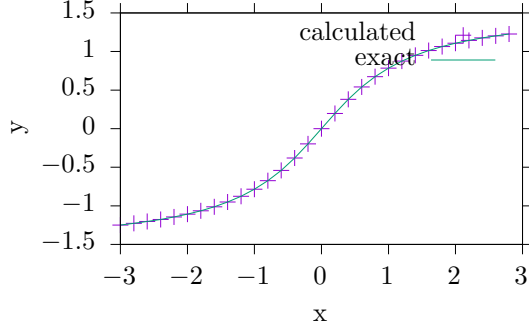


Figure 3: Comparison between the calculated modified arctangent function using differential equation found in "myarctan2.c" and the arctangent function from the math.h library.

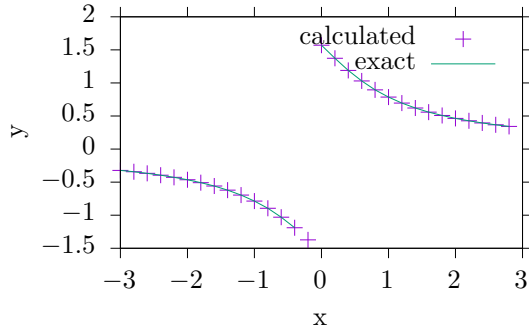


Figure 4: Comparison between the calculated modified arccotangent function using differential equation found in "myarccot2.c" and the arccot function from the math.h library, by calculation of arctan of the inverse value.

2 Implementation and Solution

The implementation is done just as in the error-function exercise, by a solution of the differential equation, as represented by equation (1) and (2). For arctan one can solve the differential equation directly, since the integration limits is from 0 to x . This can be done just as in the error-function exercise. For the arccot function, if one wants to implement the same method directly, by starting from 0, you would need to reduce equation (2). This can be done avoiding the singularity at $z = 1$, and using equation (1) by

$$\begin{aligned}
 \operatorname{arccot}(x) &= \int_x^\infty \frac{1}{z^2 + 1} dz \\
 &= \int_0^\infty \frac{1}{z^2 + 1} dz - \int_0^x \frac{1}{z^2 + 1} dz \\
 &= \arctan(\infty) - \arctan(0) - \int_0^x \frac{1}{z^2 + 1} dz \\
 &= \frac{\pi}{2} - 0 - \int_0^x \frac{1}{z^2 + 1} dz \\
 &= \frac{\pi}{2} + \int_0^x -\frac{1}{z^2 + 1} dz
 \end{aligned} \tag{4}$$

This is implemented by the well known `GSL_odeiv2.h` library. The differential equations solved is given in equation (1) and (4), and are very similar except for a change of sign and an offset. The function is included as the `sys.function`. The `sys.jacobian` is set to `NULL` as well as the `sys.params`. The dimension of the problem is only one, and the driver applied is the generally accurate `gsl_odeiv2_step_rkf45`. For the arctan we know the initial conditions $\arctan(0) = 0$, and due to equation (3) we therefore have start-condition for $\operatorname{arccot}(0) = \pi/2$. From here on, we just advance in steps defined by the driver, until we reach the desired value. Due to the fact that both of the arctan and arccot are odd functions, we can reduce the numerical calculation to positive values, by the identity of odd functions

$$\arctan(-t) = -\arctan(t), \tag{5}$$

$$\operatorname{arccot}(-t) = -\operatorname{arccot}(t). \tag{6}$$

By this method, the plots on figure 1 and 2 are made and compared to the tabular functions defined

in the `math.h`-library. Here we for the undefined `arccot` function define it using `math.h` as:

$$\operatorname{arccot}(x) = \arctan\left(\frac{1}{x}\right) \quad (7)$$

On the second two figures 3 and 4, we have implemented a check for whether the argument in the functions are above or below 1, to satisfy the condition to solve $\arctan(x)$, if $x \leq 1$, and $\operatorname{arccot}(x)$ for $x > 1$, and compensate by formula (3). There is no visible difference between the ordinary solutions 1 and 2 compared to the modified versions in figure 3 and 4. To show the differences between the two figures, I have tried to use a less accurate stepping function, and allowing the step-function to give higher errors, but the driver seems to compensate, and therefore a visible difference has not been shown here in the report. However the modified functions are denoted in the files by a 2, and give the correct results.

As an optional task, one should argue whether the ideal point to switch integration routine, should be just at the argument $x = 1$. This is simple to show, since the numerical differential equation solver for the two equations only differ by a sign, making them equally difficult to solve for the driver. Thereby since the functions respond equally to the driver, the function, which returns the lowest output must be the most accurate, since the computing accuracy is best at small numbers. Thereby as $\arctan(0) = 0$ and $\operatorname{arccot}(0) = \pi/2$, we must for small x have highest accuracy evolving \arctan . At high x , we will have \arctan approaching $\pi/2$ and according to equation (3) have arccot approaching 0, making the arccot be the most accurate. Since they both converge with exactly the same ratio, one would find the ideal point to switch integration routine, to be exactly when the two functions are equal, where they due to equation (3) both return $\pi/4$. This happens exactly at

$$\arctan(1) = \operatorname{arccot}(1) = \frac{\pi}{4} \quad (8)$$

Therefore at $x = 1$ we have the ideal place to change integration routine. This conclusion as well as the implementation of both the initial code, and the modified code ends this project.