

# Newton-Raphson Inverse

February 5, 2020

## 1 Algorithm

This code computes the inverse of a number using the Newton-Raphson formula:

$$x_{n+1} = x_n(2 - ax_n) \tag{1}$$

## 2 What to focus on

1. how COBOL implements function

## 3 Notes

This is the same code as for the example 001. You should refer to that set of notes for details. The only new file is NRreciprocalFunc.cbl. In 001 we use the file NRreciprocal.cbl to implement a "function" in cobol. In that case the "function" (actually called a subroutine) is a line with the function name followed by the body of the function. One main difference with an actual function in java or kotlin is the absence of a list of arguments, their types, etc. Is there something similar in cobol?

It turns out there is. If you check file NRreciprocalFunc.cbl, the function is now defined in a more "usual" way. In this case, we define the variables of the function and its arguments. And we actually need to pass those arguments as well (notice the slight change in the perform loop). As you can see, you still need to define the same divisions as for a full program with the addition of the linkage division that deals with the arguments passed. This might be

closer to the way functions are implemented in java and kotlin, but notice the verbosity of the code when compared to NRreciprocal.cbl.

## **4 Translation**

The translation is similar to what we saw in 001. However, pay attention to the data types. In cobol we use pictures (the PIC keyword) for the arguments. As discussed in 001, make sure to use the BigDecimal data type.