Analysis of *the Rule of 70* with MLAB

Daniel R. Kerner
Civilized Software, Inc.
12109 Heritage Park Circle
Silver Spring MD 20906
Tel.: (301)-962-3711
email: kerner@civilized.com
URL: http://www.civilized.com

Abstract

The Rule of 70 states that the time required for a fixed income investment to double in value is approximately $70/R$, where R is the fixed interest rate per compounding period. The reliability of *the Rule of 70* is analyzed. The rule is exact when $R \approx 1.9838$; the rule overestimates the doubling time when $R < 1.9838$; and the rule underestimates the doubling time when $R > 1.9838$. The value 70 is found to be the nearest multiple of 10 for the parameter a in the function a/R that best approximates, in the least squares sense, the exact doubling time. *The Rule of 70* is within 1 percent of the exact doubling time for $0 < R < 4.02353076$, within 5 percent of the exact doubling time for $0 < R < 12.3144692$, and within 10 percent of the exact doubling time for $0 < R < 22.9657819$. *The Rule of 70* is also shown to provide reasonable estimates of the time required for an investment subject to a fixed percentage loss to lose half of its value.

The value of a fixed interest rate investment with principal P, and interest rate R, after n compounding periods, is given by:

$$((1 + \tfrac{R}{100})^n)P$$

The Rule of 70 provides a simple and convienient method for estimating the time required for an investment paying a fixed rate of return to double in value. The exact time required for an investment to double in value is found by solving the following equation for n:

$$2P = ((1 + \tfrac{R}{100})^n)P$$

The solution is: $n = \frac{\log(2)}{\log(1+\frac{R}{100})}$. *The Rule of 70* states that if an investment returns an amount equal to the percentage R of the amount currently invested in each period of time over the life of the investment, then the value of the investment, comprised of principal and compounded interest, will double the initial value after approximately $(70/R)$ periods of time, i.e.

$$n = \frac{\log(2)}{\log(1+\frac{R}{100})} \approx \frac{70}{R}$$

As an example, consider an investment returning 5 percent each year. According to *the Rule of 70*, the principal and interest will double the initial principal value after $70/5 = 14$ years. A more accurate value for the doubling time for an investment returning 5 percent each year is 14.2066991 years; in this case *the Rule of 70* provides a value that is on the low side but correct to 1 part in 100.

However *the Rule of 70* is not always so accurate. As another example, consider an investment returning 100 percent each year. According to *the Rule of 70*, the principal and interest will double the initial principle after $70/100 = 0.7$ years. The actual doubling time for an investment returning 100 percent each year is one year. So here *the Rule of 70* is significantly lower than the exact doubling period.

The MLAB mathematical modeling computer program is a tool that allows the user to explore algebraic, differential equation, and implicit-function models, to name just a few applications. This paper demonstrates its use in determining the range of validity of *the Rule of 70.* The MLAB program is available for Windows, Linux, and Macintosh systems. The MLAB program presents the user with a window in which commands are typed and numerical responses from MLAB are generated. Alternately, MLAB do-files (scripts) may be constructed and executed, and reused as desired. MLAB also features a variety of graphics commands that allow the user to create another window to display graphical output. "Scripts" of MLAB commands called do-files, are runnable with the command `DO F`.

The following text shows the MLAB commands for computing the doubling time of an investment according to *the Rule of 70* and compares the result to the exact formula. (Note, comments in the lists of MLAB commands that follow that are delimited by `/*` and `*/` are ignored by MLAB.)

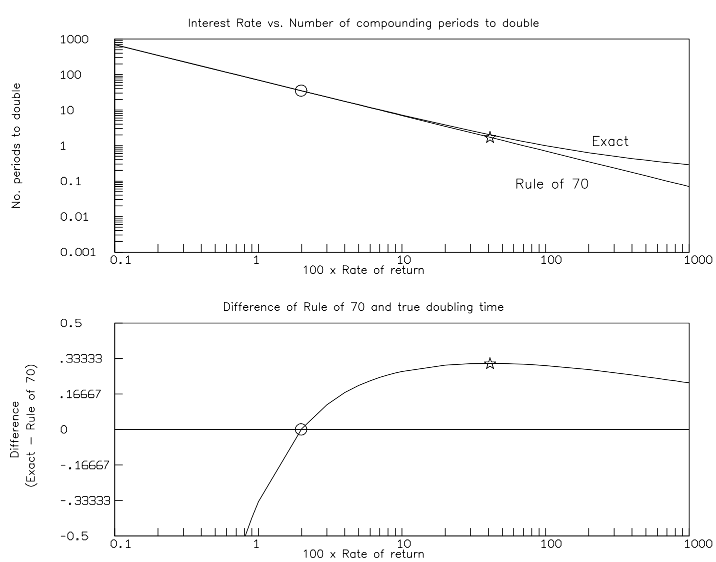
```
/* Define a function for the exact doubling time for
   a given interest rate.*/
fct d(r) = log(2)/log(1+r/100)

/* Define a function for the doubling time according
   to the Rule of 70.*/
fct e(r) = 70/r

/* Define a function for the difference between the
   Rule of 70 and exact doubling times.*/
fct f(r) = e(r)-d(r)

/* Draw graphs of the functions d(r) and e(r) in
   window w1, and the difference function d(r)-e(r)
   in window w2, over four ranges of interest rates.*/
r = (.1:.9:.1)&(1:9:1)&(10:90:10)&(100:900:100)&1000
draw points(d,r) in w1
draw points(e,r) in w1
draw points(f,r) in w2
```

With suitable commands (not shown) to make the MLAB graphical window w1 have log-log axes and the MLAB graphical window w2 have log-linear axes, the following picture is obtained:



The upper graph shows the exact and *Rule of 70* approximation of the doubling time as a function of percentage rate of return. The lower graph shows the difference between the exact and *Rule of 70* approximation of the doubling time as a function of the interest rate of return. Two features in these graphs are apparent and highlighted by a circle and a star. First, there is an interest rate at which *the Rule of 70* doubling time equals the exact doubling time. This point is where the difference curve intersects the line $y = 0$ and is marked by a circle. The numerical value of the interest rate at the point of intersection can be determined in MLAB as follows:

```
/* Define a function for the difference between the
   Rule of 70 and exact doubling expressions.*/
fct z(t) = ROOT(r,0.001,100,e(r)-d(r))
type z(0)
```

The `ROOT` command searches the interval of `r`-values from `0.001` to `100` for a value where `e(r)-d(r)` is zero. MLAB responds to these commands with:

```
 = 1.98380058
```

So the doubling time as determined by *the Rule of 70* is nearly exact if the interest rate is 1.98380058 percent. It is apparent from the lower graph that at interest rates less than 1.98380058 percent, *the Rule of 70* overestimates the exact doubling time and at interest rates larger than 1.98380058 *the Rule of 70* underestimates the exact doubling time.

The lower graph also shows that although there is no lower limit to the difference, there is an upper limit to the difference. Beyond this point, the *Rule of 70* becomes increasingly accurate as the rate of return increases. The upper limit to the difference, marked on the graphs with a star, can be found by obtaining the interest rate at which the derivative of the difference of *the Rule of 70* doubling time from the exact doubling time–with respect to interest rate, is zero. The interest rate at which the derivative is zero can be computed with the following MLAB commands:

```
/* Compute the interest rate at which the error
   is a maximum.*/
fct y(t) = ROOT(r,0.001,100,e'r(r)-d'r(r))
type y(0)
```

Note that in the MLAB language, the derivative of a function, `f(t)`, with respect to its argument, `t`, is expressed as `f't(t)`. MLAB responds to these commands with:

```
 = 41.0242057
```

This response indicates that the maximum underestimate of the doubling time according to *the Rule of 70* occurs with the interest rate equal to 41.0242057 percent.

MLAB also provides a non-linear curve fitting command which allows one to determine the value of the proportionality constant, a, in the expression $e(r) = a/r$ that has the least sum of squares error. The following series of commands determine the best value:

```
/* Define a function to fit the doubling function that
   is inversely proprotional to the rate of return.*/
fct g(r) = a/r

/* Provide an initial guess for the value of the
   parameter.*/
a = 1

/* Generate a matrix with exact data points over a
   wide range of interest rates.*/
mr = (.1:.9:.1)&(1:9:1)&(10:90:10)
mr = points(d,mr)

/* Reduce the sum of squares by adjusting
   parameter a.*/
fit (a), g to mr
```

In response to these commands, MLAB responds:

```
final parameter values
      value            error      dependency    parameter
    69.38459718    0.02400283349           0    A
2 iterations
CONVERGED
best weighted sum of squares = 2.329796e+00
```

```
weighted root mean square error = 2.993452e-01
weighted deviation fraction = 6.583745e-04
R squared = 9.999959e-01
```

This response indicates that the value of the proportionality factor that minimizes the sum of squares error between the data in the matrix `mr` and the model function `g` is 69.38459718. The nearest multiple of ten to this value is the value used in *the Rule of 70*.

We can use MLAB to find the maximum interest rate at which *the Rule of 70* is within 10, 5, and 1 percent of the exact doubling time as follows:

```
/* Compute the interest rate at which Rule of 70
   is within %10, %5, and %1  of the exact
   doubling time.*/
fct w(t) = ROOT(r,0.001,100,t-(d(r)-e(r))/e(r))
type w(.1), w(.05), w(.01)
```

MLAB responds to these commands as follows:

```
 = 22.9657819
 = 12.3144692
 = 4.02353076
```

These numbers mean that *the Rule of 70* is within:

1 percent of the actual doubling time for interest rates in the interval [0,4.02353076];
5 percent of the actual doubling time for interest rates in the interval [0,12.3144692]; and
10 percent of the actual doubling time for interest rates in the interval [0,22.9657819].

Finally, we note that the *the Rule of 70* provides reasonable estimates for the time required for an investment whose value is *diminishing* at a fixed percentage rate R to fall to half its initial value. In this case we must solve the equation:

$$\frac{P}{2} = ((1 - \frac{R}{100})^m)P$$

The solution is: $m = \frac{-\log(2)}{\log(1-\frac{R}{100})}$. The following MLAB commands and response compute a table of the halving time as determined by *the Rule of 70* and the exact halving time at given values of the interest rate.

```
* /* Define a function for the doubling time
*    (and halving time)according to the Rule of 70.*/
* fct e(r) = 70/r
*
* /* define a function for the exact halving times for a
* fixed loss of interest.*/
* fct l(r) = -log(2)/log(1-r/100)
*
* /* define a vector of interest rates */
* rv = list(.01,.02,.04,.08,.1,.2,.4,.8,1,2,4,8,10)
*
* m col 3 = e on rv
* m col 4 = l on rv
* namesw = 0
* type m

   M: a  13 by 4 matrix

 0.01   7000   6931.12523
 0.02   3500   3465.38932
 0.04   1750   1732.52135
 0.08   875    866.087356
 0.1    700    692.800549
 0.2    350    346.226901
 0.4    175    172.93999
 0.8    87.5   86.29636
 1      70     68.9675639
 2      35     34.3096185
 4      17.5   16.979748
 8      8.75   8.31295041
 10     7      6.57881348
```

Here, the left-most column of numbers are values of time, the middle column of numbers are the halving time computed with the *Rule of 70*, and the right-most column of numbers are the halving times computed from the exact formula.

Other applications of MLAB in the area of business, accounting, and finance include analyses of:

- the Dow Jones industrial average,
- the election to collect Social Security at age 65 or 70.5,
- the Black-Scholes mathematical model for pricing stock options,
- amortization schedule computations, and
- inventory replacement schedules based on survival functions.

For more information about these applications and MLAB, please visit http://www.civilized.com or contact Civilized Software at:

Civilized Software, Inc.
12109 Heritage Park Circle
Silver Spring, MD 20906-4555 USA
Tel. (301) 962–3711
Email: `csi@civilized.com`