Calculating Internal Rate of Return (IRR) in Practice using Improved Newton-Raphson Algorithm

Nerio Pascual
Technological Institute of the Philippines
Quezon City, Philippines
farbutnearyou@gmail.com

Ariel Sison
Technological Institute of the Philippines
Quezon City, Philippines
ariel.sison@eac.edu.ph

Bobby Gerardo
Western Visayas State University
Iloilo City, Philippines
bgerardo@wvsu.edu.ph

Ruji Medina
Technological Institute of the Philippines
Quezon City, Philippines
ruji.medina@tip.edu.ph

ABSTRACT

The most popular financial yardstick of investment productivity is the Internal Rate of Return (IRR). The fastest way to calculate IRR is by using iterative root-finding algorithms, the most popular of which is the Newton-Raphson algorithm. However, while the Newton-Raphson algorithm is the quickest, it often does not converge to the root if the user's guess initial value input is not close to the true root. This study aims to enhance the original Newton-Raphson technique by getting rid of the user's initial input value and by simply automatically generating a value closer to the true root, thereby avoiding the danger of non-convergence. As this novel algorithm is capable of estimating a proximate initial input of IRR, iterations are made fewer, thereby reducing runtime. The use of the modified Newton-Raphson algorithm in estimating IRR in test situations demonstrates that there is a significant decrease in the number of iterations versus that of the original algorithm, thereby causing far less runtime. Test outcomes show that the employment of the enhanced Newton-Raphson root-finding algorithm is a highly effective technique in determining IRR, thereby providing a substantially better approach in measuring investments.

KEYWORDS

Root-finding algorithm, Newton-Raphson algorithm, IRR, Convergence

1 Introduction

The most popular financial yardstick of investment productivity is the Internal Rate of Return (*IRR*), being used substantially in measuring lucrativeness of investment portfolios [5,7]. However, the *IRR* cannot be computed analytically, or by a closed-form approach, but by an iterative technique [15, 20].

Four of the most widely employed root-finding methods are bisection, secant, false position, and Newton-Raphson algorithms [1,12,18,19].

A lot of other techniques in finding roots of nonlinear problems exist; nevertheless, they have notable drawbacks, such as the required user's guess input initial value, non-convergence, complexity, slower speed, and low accuracy.

Although the Newton-Raphson technique is generally well-preferred for its fast quadratic convergence [11,19], it has unfavorable glitches, e.g., frequent divergence and division by zero [9,14,23]. However, once the method converges, it converges to a stationary point of the function [2].

This research is motivated by dealing with problems of the original Newton-Raphson method, such as its inability in converging to a true root due to the user's initial value guess input being far from the said root.

The objective of this research is to improve the Newton-Raphson approach in computing the *IRR* by 1) getting rid of the user's inputting of the initial *IRR* value, 2) increasing accuracy, 3) reducing the number of iterations, and 4) reducing runtime.

2 Related Literature Review

Given an investment plan where C_0 is the initial investment and C_i is the subsequent cash flow in period i, $i = 1, \ldots, n$, it is stated in [21] that the largest root of the nonlinear function

$$NPV(IRR) = -C_0 + \sum_{i=1}^{n} \frac{C_i}{(1+IRR)^i}$$

is the most appropriate rate of return. Searching for the solution to this continuous function of *IRR* provides several new prospects for developing helpful tools in finance and novel analytical methods. However, the said paper was not able to provide a method that estimates the largest root; rather, the method in [21] requires that every possible root should be determined first. Moreover, such research fell short of verifying its contention that the appropriate root is the largest one.

Methods in finding solutions to nonlinear functions were proposed in [22]: the Halley technique, the Newton approach, and the blending of the Newton technique, the Newton inverse scheme, and the Halley technique. Although the latter approach is a composite of three (3) different techniques, the pitfall of requiring a guess input initial value from the user still exists and is likely to cause divergence.

The research of [24] created a distinctively broad group of derivative-free *k*-point iterative techniques of optimal order of convergence $2^{(k-1)}$ using rational interpolants. However, these methods, likewise, have the drawback of demanding an initial IRR value user's guess.

Using numerous standard root-finding methods for digital

maximum power point tracking, [4] proposed a Modified Regula Falsi Method (MRFM) and employed it in photovoltaic uses. The paper claimed the method was faster than certain other methods and that divergence is eliminated. However, MRFM, likewise, requires the user's two (2) initial *IRR* values, which may lead to non-convergence.

The modified secant and exponential interpolation methods, proposed by [13], are affirmed as giving higher accuracy with more ease than the secant approach and can be used for manual estimation of *IRR* or, when needed, for computer programming. The modified secant method also gives an approximate of the uncertainty. Nevertheless, said modified secant method is complicated for numerical implementations.

The study of [6] declared that it had created an accurate mathematical tool that can approximate the *IRR*. However, the said method presumes only one value of the project's initial return and just one value for the cost, which presumption is not true in practice as there are normally more than just one cost value during the project's entire life.

In [16], Particle Swarm Optimization and Ant Colony Optimization algorithms were used in determining the *IRR*. However, these algorithms' accuracy and computing speed values can be low.

The research of [18] estimated *IRR* for a diminishing musyarakah model by employing bisection and secant approaches. However, said approaches' convergence capabilities are poor when they converge at all.

Frowning upon the Newton-Raphson technique for its weaknesses of non-convergence when dealing with nonconvex problems and its possible ill-conditioning due to division by zero, [14] developed what it called a shortened *IRR* method, assuming cash inflows are positive and constant. Though the approach appears to be simple, it is essentially complicated since the *IRR* could not be analytically solved and would still require a root-finding tool. Moreover, such a scheme supposes cash inflows to be identical and positive, which supposition is practically not true in the real world.

Believing that employing trial and error method in computing the *IRR* of complex investments using a sequential procedure may be computationally expensive, the researchers in [3] evaluated the solution by experimentation with parallel computations on graphics processing units (GPUs), regular sequential style, and a database-driven structure. While the structured query language (SQL) approach is gauged to be the quickest, it, however, is complex and costly computationally.

The research of [9] found that estimating the *IRR* is so big an information technology puzzle that they developed a fuzzy technique in attempting to crack it. However, the said study has not shown the speed and accuracy of such an approach.

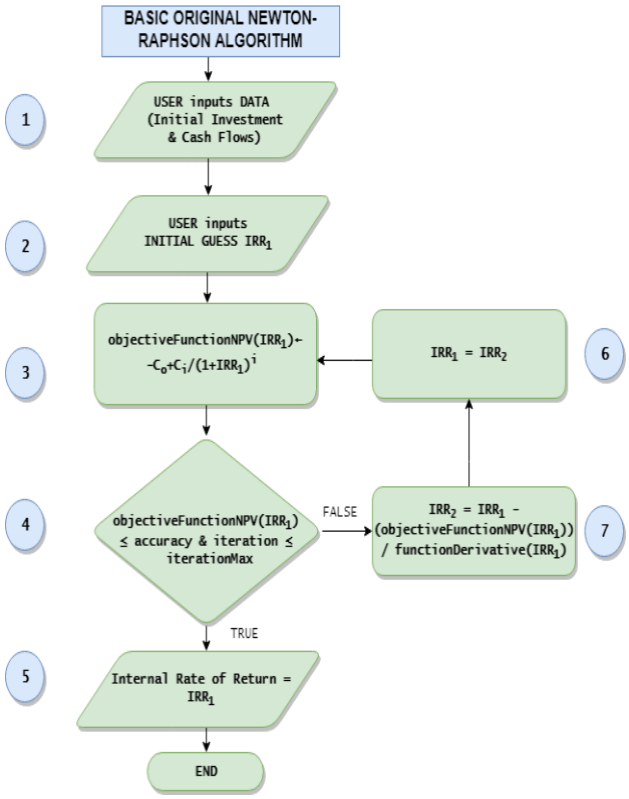
The study of [16] declared that the determination of the *IRR* is so complex that it proposed the use of an abridged closed-form method. However, this method yielded low accuracy.

In the next section, we present an alternative way of determining an initial solution that could greatly enhance the speed of computation.

3 Improved Newton-Raphson Method

This study applies the experimental approach by putting side by side the original Newton-Raphson method, as demonstrated by Figure 1, and this study's proposed improved Newton-Raphson technique, as presented in Figure 2. In the figures below, *objectiveFunctionNPV*(IRR_1), defined as *NPV*(IRR_1), may (not necessarily) be the Net Present Value of total cost, or total yield, or any monetary value which occurs at a later time. It is the Net Present Value of all cash flows discounted by *IRR*. Also, *functionDerivative*(IRR_1) is the derivative of *objectiveFunctionNPV*(IRR_1) .

Figure 1. Original Newton-Raphson Algorithm



At the true root, *objectiveFunctionNPV*(IRR_1) is assumed to be zero (0). The objective then is for *objectiveFunctionNPV*(IRR_1) to reach zero or, at least, an acceptable tolerance level that is close to zero [8,13].

Observe that, in Step 2 of the original algorithm, the user inputs his guess of the initial IRR_1.

In the improved Newton-Raphson algorithm, Step 2 is modified (see Figure 2) by replacing the user's guess input of initial IRR_1 by the following formula:

$$IRR_1 \leftarrow \left(\frac{\sum_{i=1}^{n} C_i}{C_0}\right)^{\frac{1}{\frac{n-1}{2}+1}} - 1,$$

That is, the initial input for IRR_1 is

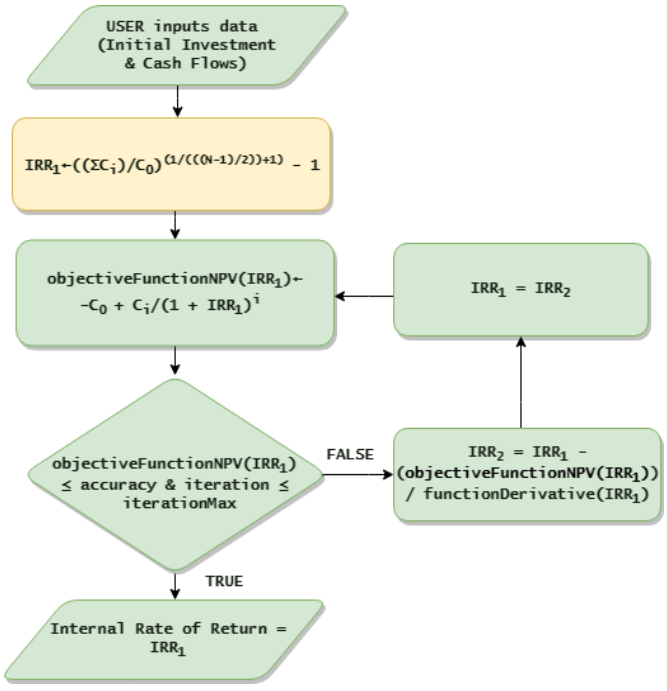
$$IRR_1^0 = \left(\frac{\sum_{i=1}^{n} C_i}{C_0}\right)^{\frac{2}{n+1}} - 1.$$

The said formula is estimated from the future value formula $FV = PV(1+r)^n$ where

$$FV = \sum_{i=1}^{n} C_i, \quad PV = C_0, \quad r = IRR,$$

and we take $n \leftarrow \frac{n-1}{2} + 1$.The formula $\frac{n-1}{2} + 1$ is the midpoint of the time periods of the respective cash flows C_1, C_2, …, C_n. To be able to use the above formula, there has to be considered only one time period n*, and the midpoint of the several time periods for C_i, where i= 1, …, n, is a good representative value of such time period n*. In the said formula, the initial value of IRR_1 depends on the cash flows C_0, C_1, …, C_n. This IRR_1 is proximate to the true root.

Figure 2. Improved Newton-Raphson Algorithm



The enhancement, using the said formula, is made in order to give an initial IRR_1 value that is close to the true root, since a value not close to the true root may lead to divergence of the method. Trials show that the percentage of the difference between the initial IRR and the final *IRR* over said final IRR, i.e.,

$$Difference = \frac{|IRR_1^0 - IRR_1^f|}{IRR_1^f}$$

is significantly small. The final *IRR* is the approximate root of *NPV*(*IRR*). A categorical determination of what is considered close to the true solution is not an objective of this study.

A value that is not sufficiently close to the true root causes divergence. The improved type has the capability of not requiring the user to input initial IRR_1, which often is not proximate to the true root, thereby causing divergence. It simply automatically computes the said initial IRR_1 from the inputted cash flows, C_0, C_1, …, C_n, as stated above. This automatically computed initial IRR_1 is close to the true root, as attested by numerical convergence of the method.

The proposed initial *IRR* is different from the derivations of R by Shestopalloff and Shestopaloff in that said derivations did not use the midpoint of the periods of the respective cash flows C_1, C_2, C_3, …, C_n [21]. On the other hand, Shestopalloff and Shestopaloff used too many (at least 19) formulas to approximate *IRR*. Moreover, finding the largest *IRR* as the true *IRR* is impossible inasmuch as it requires testing every possible *IRR*s (which may be infinite in number) to find out whether such *IRR* is a root of *NPV*, that is, an input value that produces an output of *NPV*(*IRR*) = 0. Worse is not all *IRR* values are realistic, for, although they may result to a zero *NPV*, they are not acceptable by common practical business sense.

4 Numerical Experiments

4.1 Result of Automatic Initial Value Computation

In this section, we demonstrate the use of the proposed *IRR* in the improved Newton-Raphson method without the user inputting a guess *IRR*.

The enhanced method employs the full step in generating the iterative solution using the Newton-Raphson method, i.e., in the general iteration

$$IRR_{\{k+1\}} = IRR_k - \alpha_k \left(\frac{d}{dIRR} NPV(IRR_k)\right)^{-1} NPV(IRR_k),$$

the step length $\alpha_k = 1$, for all $k = 0,1,2,\ldots$.

As a test case in this research, consider the data presented in Table 1, entitled "X Corporation Cash Flows", from a certain Philipine company in existence for 29 years as of 2018. The cash flows are C_0 = - P6M; C_1, C_2,C_3 = - P3M; C_4,...,C_9 = P5M; C_{10},...,C_{13} = P6.6M; C_{14},...,C_{19} = P10.6M; C_{20},...,C_{24} = P19M; C_{25},...,C_{29} = P31M. Note that a negative yearly cash flow indicates that the overall cash inflow is less than the sum of cash outflows for that year. This situation occurs during first few years the company was operating.

The automatically calculated initial IRR_1^0 is 0.3141. Applying the improved Newton-Raphson approach, the method converged to the final IRR 0.257591358915355 (or 25.7591358915355%) in 5 iterations (see Table 3). This result of the modified method is very close to the true root, at a distance of 5.96 x 10^{-8}. Further, the

closeness of IRR_1^0 to the final *IRR* is *Difference*=21.93%. In comparison, when the user input a guessed initial IRR_1, e.g., 0.65, 0.5, or 0.35, the method used more iterations, and consequently more runtime, to converge to a solution There were even cases of non-termination of the method. Table 2 displays results of this approach. In the table, *iter* stands for number of iterations to converge and *error* is the distance of the obtained final IRR_1 from the true IRR_1. In the table, NA stands for not applicable, and NaN stands for not-a-number which is usually the case when there is division by zero.

When the simple interest method is applied, a method prevalently used in many business establishments, the resulting *IRR* is 204.02%, resulting in an unacceptably high error of 692.04% with respect to the true *IRR*.

Table 1. X Corporation Cash Flows

Year	*Cash Flow (in PHP)*	*Year*	*Cash Flow (in PHP)*
0	-6,000,000.00	17	10,600,000.00
1	-3,000,000.00	18	10,600,000.00
2	-3,000,000.00	19	10,600,000.00
3	-3,000,000.00	20	19,000,000.00
4	5,000,000.00	21	19,000,000.00
5	5,000,000.00	22	19,000,000.00
6	5,000,000.00	23	19,000,000.00
7	5,000,000.00	24	19,000,000.00
8	5,000,000.00	25	31,000,000.00
9	5,000,000.00	26	31,000,000.00
10	6,600,000.00	27	31,000,000.00
11	6,600,000.00	28	31,000,000.00
12	6,600,000.00	29	31,000,000.00
13	6,600,000.00	IRR=	25.7591358915355%
14	10,600,000.00	Simple Interest=	204.02%
15	10,600,000.00	Simple Interest Method Error %=	692.04%
16	10,600,000.00		

Another test *case*, particularly an Installment Plan of *a* Chevrolet Trailblazer, consists of cash flows C_0 = - P1,438,888.00 and C_1,…, C_{60} = P32,269.00. For this case, the simple interest approach presents an interest rate of just 0.58%. However, the accurate IRR figure is 0.0102992630681530 (or 1.02992630681530%). The simple interest method result is inaccurate by at least 40%.

On the other hand, the modified method yielded a result that is closer to the true *IRR*. In Table 3, the final *IRR* of our approach is off the true *IRR* only by 9.31 x 10^{-10}.

Table 2. Original Newton-Raphson Algorithm

Test case	*Initial Guess*	*Final IRR*	*Iter*	*Runtime (in ms)*	*Error*	*Remarks*
X Corp.	-1	NA	NA	NA	NaN	Error returned
	1	NA	No convergence in 1000	NA	NA	No convergence
	0.35	0.257591358915356	6	210	2.42143869400024E-08	Slow convergence
	0.25	0.257591358912485	3	48	2.31713056564331E-04	Quick convergence but lower accuracy
	0.5	0.257591358915355	22	294.6666666667	1.12690031528472E-07	Slow convergence
	0.65	0.257591358915356	36	416	4.65661287307739E-09	Slow convergence
	0.75	NA	No convergence in 1000	NA	NA	No convergence
Chevy Trail-blazer Install-ment Purchase	-1	NaN	NA	NA	NA	NaN
	1	NA	No convergence in 1000	NA	NA	No convergence
	0.5	NA	No convergence in 1000	NA	NA	No convergence
	0.25	NA	No convergence in 1000	NA	NA	No convergence
	0.15	0.010299263	76	1674	1.28E-04	Slow convergence

Using the original method, with user-guessed input initial *IRR*, the method converged slowly, if at all. The result can also be far off the true *IRR*. See Table 2. In the modified method, the initial IRR_1^0 is 0.009779500216542. Table 3 below shows the resulting final rate

of 0.010299263068153, obtained in 3 iterations only. Again, the numerical result shows the closeness of the initial solution with the final rate, with *Difference*=5.05%.

Table 3. Enhanced Newton-Raphson Algorithm

Test Case	*Initial Guess*	*IRR*	*Iter*	*Runtime (in ms)*	*Error*	*Remarks*
X Corp.	Guess Not Needed	0.257591358915355	5	133	5.96046447753906E-08	Quick convergence
Chevy Trail-blazer Installment Purchase	Guess Not Needed	0.010299263068153	3	82.6666666667	9.31322574615478E-10	Quick convergence

As observed in these two test cases, the improved Newton-Raphson algorithm always converges to the true root. However, this study does not make a general claim of convergence for all test problems.

5 Conclusions and Recommendation

The improved Newton-Raphson algorithm gives good results that are close to the true solution without the need of a user's initial guessed values for the rate.

Likewise, the modified algorithm improves on the speed of the original method with user input, since the proposed automatically generated IRR_1^0 value is close enough to the final rate. The runtime is significantly decreased as less iterations are needed to reach convergence.

This study recommends further experimentation of the modified algorithm in other test scenarios, for instance, in investment decision making and in determining realistic interest rates in compliance with the Truth in Lending Act. Further research may also be made where a variable step length is considered.

Finally, this study recommends future in-depth theoretical analysis on the convergence of this enhanced method.

REFERENCES

[1] A. G. Ahmad, "Comparative Study of Bisection and Newton-Rhapson Methods of Root-Finding Problems," 121–129, 2015.

[2] D. Bertsekas, "Nonlinear Programming,"2nd ed, Athena Scientific, 2003

[3] R. Casturi, "Design and Development of an Efficient Calculation Framework for Internal Rate of Return (IRR) of a Fixed Income Portfolio with SIMD Architecture .," *Int. J. Emerg. Technol. Adv. Eng.*, 4 (9):, 2014.

[4] S. Chun and A. Kwasinski, "Analysis of classical root-finding methods applied to digital maximum power point tracking for sustainable photovoltaic energy generation," *IEEE Trans. Power Electron.*, 26 (12): 3730–3743, 2011.

[5] R. Costello and A. Pecher, *Economics of WECs*, (1):. 2017.

[6] Y. El-Tahir and D. El-Otaibi, "Internal Rate of Return: A suggested Alternative Formula and its Macro-economics Implications," *Jounal Am. Sci.*, 10 (11): 216–221, 2014.

[7] Eric B. Storey, "(Keep score: Using internal rate of return to score investments: 'and the winner is...,'" *J. Prop. Manag.*, 6 (May): 6–7, 2016.

[8] S. R. Faisalabad, "Net Present Value is better than Internal Rate of Return

[9] A. Gholami and H. S. Aghamiry, "Iteratively re-weighted and refined least squares algorithm for robust inversion of geophysical data," *Geophys. Prospect.*, 2017.

[10] V. B. Gisin and E. S. Volkova, "Internal Rate of Return of Investment Projects with Fuzzy Interactive Payments," 0 731–733, 2017.

[11] Y. Liang, Z. Shi, and P. W. Chung, "A Hessian-free Newton–Raphson method for the configuration of physics systems featured by numerically asymmetric force field," *Math. Comput. Simul.*, 140 1–23, 2017.

[12] D. Mancusi and A. Zoia, "Chaos in eigenvalue search methods," *Ann. Nucl. Energy*, 112 354–363, 2018.

[13] J. Moten and C. Thron, "Improvements on secant method for estimating Internal Rate of Return (IRR)," (September 2017):, 2013.

[14] M. M. Mujahed and E. E. Elshareif, "Internal Rate of Return (IRR): A New Proposed Approach," *Benlamri R., Sparer M. Leadership, Innov. Entrep. as Driv. Forces Glob. Econ. Springer Proc. Bus. Econ. Springer, Cham*, 1–9, 2017.

[15] M. J. P. Nijmeijer, "A parallel root-finding algorithm," *LMS J. Comput. Math.*, 18 (01): 713–729, 2015.

[16] Z. Qiao and H. Zhang, "Research on estimation methods of internal rate of return in hydraulic engineering project," *2nd Int. Conf. Inf. Eng. Comput. Sci. - Proceedings, ICIECS 2010*, (2):, 2010.

[17] A. de S. Rangel, J. C. de S. Santos, and J. R. F. Savoia, "Modified Profitability Index and Internal Rate of Return," *J. Int. Bus. Econ.*, 4 (2): 13–18, 2016.

[18] S. Z. M. Ruslan and M. M. Jaffar, "Application of numerical method in calculating the internal rate of return of joint venture investment using diminishing musyarakah model," 020007 020007, 2017.

[19] M. Salimi, T. Lotfi, S. Sharifi, and S. Siegmund, "Optimal Newton–Secant like methods without memory for solving nonlinear equations with its dynamics," *Int. J. Comput. Math.*, 94 (9): 1759–1777, 2017.

[20] A. A. Sangah, A. A. Shaikh, and S. F. Shah, "Comparative Study of Existing Bracketing Methods with Modified Bracketing Algorithm for Solving Nonlinear ...," *SINDH Univ. Res. J. (SCIENCE Ser.*, (October):, 2016.

[21] Y. Shestopaloff and A. Shestopaloff, "Choosing the Right solution of iRR equation to measure investment success," (D):, 2013.

[22] B. P. Silalahi, R. Laila, and I. S. Sitanggang, "A combination method for solving nonlinear equations," *J. Phys. Conf. Ser.*, 755 011001, 2017.

[23] T. Yamamoto, "Historical developments in convergence analysis for Newton's and Newton-like methods," *J. Comput. Appl. Math.*, 2000.

[24] F. Zafar, N. Yasmin, S. Akram, and M. U. D. Junjua, "A general class of derivative free optimal root finding methods based on rational interpolation," *Sci. World J.*, 2015 (3):, 2015.