Java Based Computer Algorithms for the Solution of a Business Mathematics Model

Chinedu, A. D. and Adeoye, A. B
Department of Mathematics, Statistics and Computer Science,
College of Science and Technology, Kaduna Polytechnic,
Kaduna, Nigeri

***Abstract*—A novel approach is proposed as a framework for working out uncertainties associated with decisions between the choices of leasing and procurement of capital assets in a manufacturing industry. The mathematical concept of the tool is discussed while the technique adopted is much simpler to implement and initialize. The codes were developed in Java-programming language and text-run and executed on a computer system running on Windows 7 operating system. This was done in order to solve a model that illustrates a case study in actuarial mathematics. Meanwhile the solution obtained proves to be stable and proffers to suit the growing frenzy for software for similar recurring cases in business. In addition, it speeds up the computational results. The results obtained using the empirical method is compared with the output and adjudged excellent in terms of accuracy and adoption.**

Keywords*—Actuarial Mathematics; Java-programming Language; Leasing; Procurement; Capital Assets; Uncertainties

I. INTRODUCTION

This paper is directly geared towards presenting mathematics as a veritable tool in financial investments of which an average businessman must know the fundamental principles involved in the use of these tools[1]. However, the business man ought not only to have the traditional personal qualities expected of a leader and expert in the field, but he must also know the fundamental principles involved in the use of the most modern tools for financial management[2]. Actuarial mathematics, or rather mathematics of Finance is amongst these tools. This branch of mathematics is certainly not new, but with the proliferation and increased capacity of financial institutions today, it becomes very pertinent that the tools of the trade be advanced to match the growing trend [3]. An organization needs not concern herself with the ways in which a certain model or formula is obtained, but must know as much as possible about the use of them in terms of their adaptability to his problems [2]. Hence an optimum advantage of this tool could be reached, when there are the possibilities of achieving a faster and more accurate solution whenever it is necessary [4]. Secondly, office automation has been in vogue among business communities. Ignorantly the acquisition of computers of all sizes and capacity seem to be used to measure the success of an establishment only. Yet, hardly do they know that computers can be used extensively for solving problems in this field of mathematics. Therefore, it becomes imperative to make a tentative survey of what is essential for an organization to know how to write or use computer programming in solving problems of Actuarial mathematics. Hence, part of this endeavor is to prepare and present a model with computer program package that handles some of the inherent problems. This will in go a long way to make the operations faster, more accurate and more reliable. A business model is a sustainable way of doing business, and sustainability stresses the ambition to survive over time and create a successful, perhaps even profitable, entity in the long run [5].

II. PROBLEM STATEMENT

The widespread use of computers in business organizations creates needs for some of mathematical problems associated with these organizations to be solved with aid of computers. Moreover some of these problems are routine and their solutions, if obtained, can easily be stored in a computer memory. With the above scenario in mind, several questions arise as regards the use of computer to solve such problems in Actuarial mathematics via computer programs in order to achieve a faster and more accurate solution. This paper however, seeks to provide an answer to similar problem by developing and test-running a computer algorithm for solution to a model case, written in java programming language.

III. METHOD OF ATTACK

Programming is not only coding. Primarily it implies structuring of the solution to a problem and then refines the solution step by step [6]. On the first instance, the applications of mathematical and experimental techniques are facilitated as compared to manual procedure when computer facilities are used. Furthermore, it will be instructive to outline the complete process of setting up a representative technical problem for computer solution to see just what a person does and what a computer does. This is where programming helps since computer cannot follow direct orders. Effective answers to prevailing queries in actuarial mathematics make it necessary to achieve a precise and unambiguous statement of exactly what we want the computer to do in terms of operations of which it capable[7]. Meanwhile, the stability of the computer algorithms depends on the language used and clearly defining what is to be done. The computer language imposes restrictions of its own in terms of what kind of orders it can interpret and execute. Besides, there is too much likelihood to make mistake in programming. The mistakes must be located and corrected. While the program must be thoroughly tested to prove that it actually does what the writer meant to do. Another thing that matters in ensuring stability is the correct interpretation of the output or result from the computer. The computer is faster and more accurate than a human being, but it cannot decide how to

proceed or what to do with result. On the general method of attack, there are different methods or rather several procedures involved in solving a problem with a computer. Moreover, in placing today's computing power in perspective, there is much more to solving a problem with a computer than the part the computer does [8]. Insolving a problem, you have to define the problem itself. This is necessary since the computer cannot do it on its own. Secondly, we formulate the mathematical description of the process under study. It is also appropriate to use numerical analysis if the need arises. The algorithm is then formulated in agraphical form, which is the flowchart. The program checkout or debugging is carried out in order to minimize the chances of in putting garbage into the computer. Finally, the program can be combined with problem data and run. From here, the computer produces results or output of which should be interpreted correctly. For this paper, Java programming language is used for the codes.

IV. THE TYPICAL PROBLEM

This paper looks into the model problem as proposed by [1] and attempts to solve same using a different programming approach. Suppose a company is considering either buying machinery for $\#\boldsymbol{x}.00$ or leasing it for $\#\boldsymbol{y}.00$ per month. Assume that money is worth $\boldsymbol{r}\%$ for $\boldsymbol{jth}$ period of time may be; annually, semiannually, quarterly or monthly and the life of the machinery is $\boldsymbol{M}$ years, after which the salvage value becomes $\#\boldsymbol{z}.00$. suppose the company could purchase a maintenance contract for $\#\boldsymbol{k}.00$ per month. Then advise the company on whether they should buy or lease the machinery.

Let the interest conversion period be $\boldsymbol{j}$ then the total number of the interest periods is $\boldsymbol{n} = \boldsymbol{m_j}$ where **m** is the asset's life span. Then if $\boldsymbol{i}\%$ is the interest rate in decimal form therefore $\boldsymbol{i} = \frac{\boldsymbol{r}}{\boldsymbol{100}}$. Here we have $\boldsymbol{y} < z < x$and $\boldsymbol{m}, \boldsymbol{n} > 0$, but if $\boldsymbol{y} \geq \boldsymbol{x}$, then the company will just buy the machinery for $\#\boldsymbol{x}.00$ instead of leasing it which will be costlier.

V. DERIVATION OF THE MODEL

To solve the problem, the present value, $\#\boldsymbol{z}$ which is to be received at the end of $\boldsymbol{m}$ years and deduct the result from $\boldsymbol{x}$. Hence this is denoted by $\boldsymbol{p_c}$. The formula for finding present value at a compound interest is

$$p = A(1+i)^{-n} \qquad (1)$$

And subtract the result from $\boldsymbol{x}$ to get the real present value, which represents the present value of the cost of owing the machinery. Hence we have;

$$p_c = x - \left\{z\left(1+\frac{r}{100}\right)^{-n}\right\} \qquad (2)$$

$$p_c = x - z(1+i)^{-n} = x - \left\{\frac{z}{(1+i)^n}\right\} \qquad (3)$$

The present value of the rent $\boldsymbol{R_p}$ is for $\boldsymbol{M}$ years. However, the formula for the annuity does not include a payment at the beginning of the term, and then the rental price $\boldsymbol{R_p}$would be

$$R_p = y + y\left\{\frac{(1+i)^{n-1}-1}{(1+i)^{n-1}}\right\} \qquad (4)$$

$$R_p = y\left\{\frac{1+\left((1+i)^{n-1}\right)-1}{i(1-i)^{n-1}}\right\} \qquad (5)$$

Let $\boldsymbol{M_c}$ be the maintenance cost of the machinery which may also be included in the rental price for the same period of time $\boldsymbol{M}$ years. Suppose the company could purchase a maintenance contract or other miscellaneous expenses for servicing the machinery then the present value for $\boldsymbol{M_c}$ would be;

$$\boldsymbol{M_c} = K + K\left\{\frac{\left((1+i)^{n-1}\right)-1}{i(1-i)^{n-1}}\right\} \qquad (6)$$

$$\boldsymbol{M_c} = K\left\{\frac{1+\left((1+i)^{n-1}\right)-1}{i(1-i)^{n-1}}\right\} \qquad (7)$$

K may be found the use of the formula for depreciation.

Generally we have;$P_c + M_c = T_c$ which is the total cost of buying and maintaining the machinery.

$$T_c = P_c + M_c$$

$$= \left\{x - \left\{\frac{z}{(1+i)^n}\right\}\right\} + \left\{K\left\{\frac{1+\left((1+i)^{n-1}\right)-1}{i(1-i)^{n-1}}\right\}\right\} \qquad (8)$$

Then the following conclusions are drawn:

1) If $T_c < R_p$ then the company would be advised to buy the machinery.

2) If $T_c > R_p$ then the company would be advised to lease the machinery.

3) *If $T_c = R_p$ then the company can take any of the options since they are equal.*

VI. APPLICATION OF THE MODEL

Suppose the Company considers buying machinery for Ninety Thousand Pounds (£90, 000. 00) or lease it for Three Thousand Pounds (£3, 000. 00) per month. Assume that the money is worth 12% compounded monthly and the life of the Machinery is Five (5) years, after which time the salvage value will be Twenty Thousand Pounds (£20, 000. 00). Suppose the Company could purchase a maintenance contract for one thousand Pounds (£1, 000. 00). Then advise the company on whether they should buy or lease the machinery.

VII. SOLUTION TO THE REAL-LIFE CASE

Finding the present value of Twenty Thousand Pounds (£20, 000. 00) which is to be received in five years, $n = 5 \times 12 = 60$, since the money is worth 12% compounded monthly then

$$r = \frac{12}{12}\% = 1\%\ per\ month.$$

Then $i = \frac{1}{100} = 0.01$. Using equation (1) gives;

$$P_c = £20,000\ (1.01)^{-60} = £11,008$$

Meanwhile, the difference of the cost and leasing of the machinery which is obtained thus,

$$£90,000 - £11,008 = £78,991$$

represents the present value of the cost of owning the machinery. Using equation (4) or (5) to find the present value of the rent for five years, the$\boldsymbol{R}_p$, would be;

$$\boldsymbol{R}_p = \pounds 3,000 + \pounds 3,000 \left\{\frac{(1+0.1)^{59}-1}{(0.01)(1+0.01)^{59}}\right\}$$

$$= \pounds 3,000 + \pounds 3,000(44.404587) = \pounds 136,213.76$$

This is in line since the formula for annuity does not include a payment at the beginning of the term. This certainly seems to indicate that the company should buy the machinery. This does not however consider other factors such as maintenance, which would be included in the rental price $\boldsymbol{R}_p$. Suppose the Company could purchase a maintenance contract for $\pounds 1,000$ per month. Then the present value of the maintenance contract $\boldsymbol{M}_c$ would be;

$$\boldsymbol{M}_c = \pounds 1,000 + \pounds 1,000 \left\{\frac{(1+0.1)^{59}-1}{(0.01)(1+0.01)^{59}}\right\}$$

$$= \pounds 1,000 + \pounds 1,000\,(44.404587) = \pounds 45,404.59$$

Therefore, $P_c + M_c = T_c$ which is the total cost of buying and maintaining the machinery, gives;

$$\boldsymbol{T}_c = \pounds 78,991 + \pounds 45,404.59 = \pounds 124,395.59$$

Therefore $\pounds 124,395.59 < \pounds 136,213.76$, which is translated as$T_c < R_p$, the company would be advised to buy the machinery.

VIII. Computer Algorithm Using Java Programming Language

To compute is to determine by mathematics which does not generally depend on the programming language [9]. The point is that the algebraic expressions are directed towards providing a few specific answers to the posed questions. However, business programs tend to be oriented towards reading or accessing a great deal of data for processing information. Hence the use of computer for solving and processing bulky data is justified if the solutions must be repeated number of times. More so, the computers are capable of performing such bulky calculations/information rapidly and accurate the manual procedure, if solutions require a large amount of storage and the problem solving process can facilitate a clearer understanding of the given problem. Generally, these instances outlined above can also be identified as the advantages of computers methods of solution to that of the manual procedures.

The world is fast evolving especially in the field of computers and computing. The development of applications has also evolved along these lines using sophisticated programming languages, one of which is Java[10]. Java is a general purpose, object oriented language that runs on billions of computers and mobile devices (cell phones, smart phones and hand held devices) worldwide. Java is used in a wide spectrum of applications and it has three different editions, namely, Java Standard Edition 7 (Java SE 7) which was employed in developing the program for this paper; Java Enterprise Edition (Java EE) which is geared towards the development of large-scale, distributed networking applications and web-based applications[11]. The Java Micro Edition (Java ME) is geared towards developing applications for small, memory-constrained devices such as BlackBerry smart phones, Google's Android operating system – used on numerous smartphones, tablets (small, lightweight mobile computers with touch screens)[12]. Java has revolutionized how things work and will continue to do so for many more years to come. Java enables the development of applications that mimicked how things "objects" exists in the real world thus making it more natural and easier to program. Importantly, Java is also an open source development application.

IX. Discussion

This paper which involves vital formulas essential for solving financial problems that often arise in the business and industries was diligently treated. Software codes to solve the model problem was developed in java SE7programming language, adopted and applied satisfactorily in the solution of the model problem.

The user interface was also designed using NetBeans IDE (Integrated Development Environment) that simplified and made easier the design process by generating codes[12].While graphical objects were drag and dropped (most of these codes as developed by the IDE are not reproduced in this paper because of space constraints). The program was executed on a computer system running on Windows 7 operating system. Using computers definitely improves the speed, accuracy, reliability of the solutions developed. The output is in good agreement with the results obtained using the empirical method. Thus it is a good approach to achieving feasible and stable solutions to management problems in business and industries. Emphatically, using computer facilitates a faster means of solving challenging problems generally.

X. Conclusion

This paper brings to limelight, the importance of adopting mathematical models and computer programming in solving problems in businesses and industries. Actuarial mathematics as a veritable tool has been adjudged excellent for good and meticulous fiscal appropriations in business and industries. More so, Java programming language proves in this research to be the most friendly and versatile in use. These have justified that both tools are of great benefit to business and industry if they are fully utilized, proving that successes do not always depend on just plain luck.

References

[1] Chinedu, A. D. and Anumba, J. U. (2012). A Mathematical Model and Computer Algorithm for Solution of Problem Using D-Base IV Programming Language. (Standardizer of the Nigerian Academics, ISSN 0189-8655, Volume 8 (1) pp (146 – 154).

[2] Ching, W. and Ng, M. K. (2006). Markov Chains: Models, Algorithms and Applications. Springer Science + Business media Inc., New York, NY 10013 USA.

[3] Cvitanic, J. and Zapatero, F. (2004) Introduction to the Economics and mathematics of Financial Markets. The Massachusetts Institute of Technology (MIT) Press Cambridge, London, England.

[4] Peers, S. (2005). Business Mathematics: Revised Edition Relevant for 2005/2006 Computer Based assessment Certificate Level. CIMA'S Official Study System, (CIMA Publishing) Elsevier Inc., Amsterdam.

[5] Nielsen, C. and Lund, M. (2012). Business Models: Networking, Innovating and globalizing. Christian Nielsen, Morten Lund (Eds.) and

bookboon.com (Ventus Publishing Aps) ISBN 978-87-403-0179-3, www.bookboon.com.

[6] Backman, K. (2012). Structured Programming with C++. KjellBackman and Ventures Publishing ApS, ISBN 978-67-403-0099-4. www.bookboon.com.

[7] Duffy, D. J. (2006). Finite Difference Methods in Financial Engineering, A Partial Differential Equation Approach. John Wiley and Sons Ltd., England.

[8] Focardi, S. M. and Fabozzi, F. J., (2004). The Mathematics of Financial Modeling and Investment Management. John Wiley and Sons Inc., New Jersey.

[9] Schildt, H. (2012). Java, A Beginner's Guide, Fifth Edition. (pp: 2-3) McGraw Hill

[10] Sierra, K. and Bates, B. (2005). Head First Java, Second Edition, (pp: 400-418) Addison-Wesley.

[11] Amber, S. (2005).The Object Primer: Agile Model-Driven Development with UML 2.0, Third Edition, New York: Cambridge University Press.

[12] Deitel, P. and Deitel, H. (2012).Java How to Program, Eighth Edition. (pp 3, 555-624) England: Pearson Education Limited.

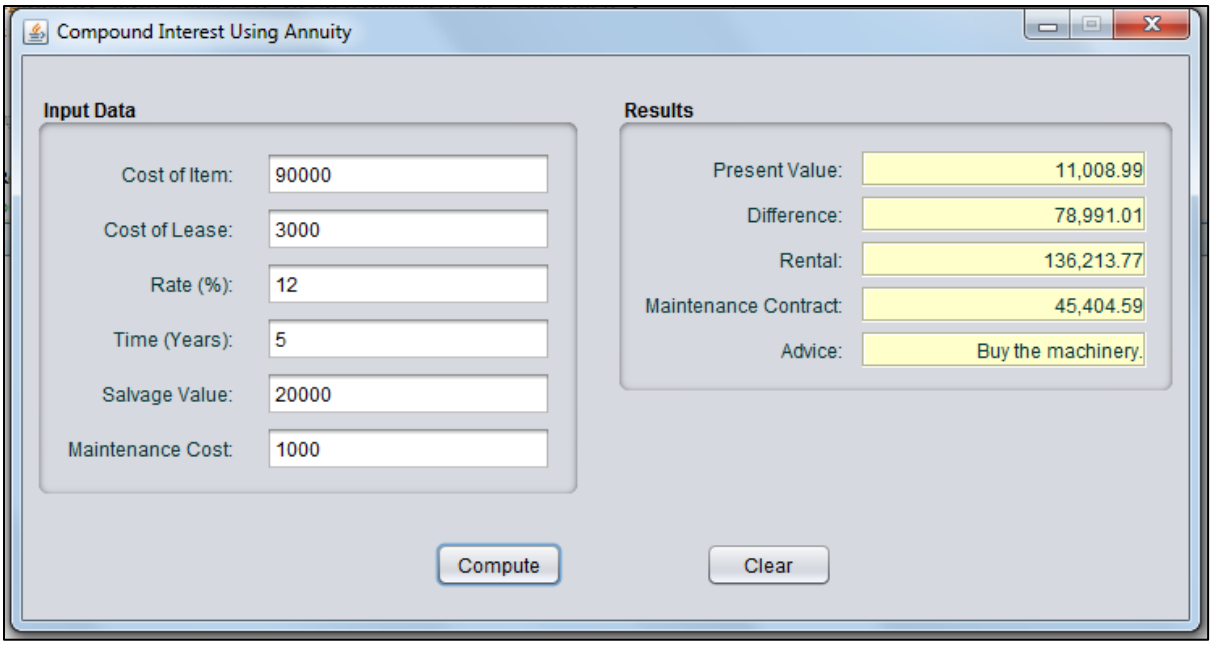

Fig. 1. Sample output

Appendix

```
/*
 * CIAnnuityJFrame.java
 */
packageedu.dele.chinedu;
importjavax.swing.JOptionPane;

/**
 *
 * @author CHINEDU and ADEOYE
 */
public class CIAnnuityJFrame extends javax.swing.JFrame
{
private double itemCost;
private double leaseCost;
private double rate;
private double salvageValue;
private double time;
private double maintCost;
private double presentValue;
private double rental;
private double tc;
private double maintContract;
private double p;
private double difference;

    /**
     * Creates new form CIAnnuityJFrame
     */
publicCIAnnuityJFrame() {
initComponents();
    }
public   void initComponents() {
computeJButton.setText("Compute");
computeJButton.addActionListener(new
java.awt.event.ActionListener() {
public void
actionPerformed(java.awt.event.ActionEventevt) {
computeJButtonActionPerformed(evt);
            }
        });

clearJButton.setText("Clear");
clearJButton.addActionListener(new
java.awt.event.ActionListener() {
public void
actionPerformed(java.awt.event.ActionEventevt) {
clearJButtonActionPerformed(evt);
```

```
        }
      });

pack();
   } // end method initComponents

private void
computeJButtonActionPerformed(java.awt.event.ActionEve
ntevt) {
compute();
displayResults();
   }

private void
clearJButtonActionPerformed(java.awt.event.ActionEvente
vt) {
clear();
   }

   /**
    * @paramargs the command line arguments
    */
public static void main(String args[]) {
      /* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
public void run() {
newCIAnnuityJFrame().setVisible(true);
        }
      });
   }
   // perform conputations
public void compute() {
try {
        // retrieve user input
itemCost = Double.parseDouble( itemCostJField.getText()
);
leaseCost = Double.parseDouble( leaseCostJField.getText()
);
rate = Double.parseDouble( rateJField.getText() );
time = Double.parseDouble( timeJField.getText() );
salvageValue = Double.parseDouble(
salvageValueJField.getText() );
maintCost = Double.parseDouble(
maintCostJField.getText() );

        // compute compound interest using annuity
double i = ( ( rate / 12 ) / 100 );
double n = 12 * time;
doublenn = n - 1;
double ii = i + 1;
double a = Math.pow( ii, nn ) - 1;
double b = i * Math.pow( ii, nn );
double c = a / b;

presentValue = salvageValue * Math.pow( ii, -n );
difference = itemCost - presentValue;
rental = ( leaseCost + ( leaseCost * c ) );
maintContract = maintCost +  ( maintCost * c );
tc = maintContract + presentValue;

}  // end try
catch(NumberFormatExceptionnfmtEx ) {
JOptionPane.showMessageDialog( null, "Please check the
values" +
" inputted. ONLY numeric values\nshould be inputted",
                "Data Entry Error",
JOptionPane.ERROR_MESSAGE );
}  // end catch

}  // end method compute

public void displayResults() {
presentValueJLabel.setText(String.format( "%,.2f",
presentValue) );
differenceJLabel.setText(String.format( "%,.2f", difference )
);
rentalJLabel.setText(String.format( "%,.2f", rental ) );
maintContractJLabel.setText(String.format( "%,.2f",
maintContract ));

      String msg = "";
if(tc< rental ) {
msg = "Buy the machinery.";
}  // endif
else if( tc> rental ) {
msg = "Rent the machinery";
}  // end else if
else
msg = "Take any option";
adviceJLabel.setText(msg );
}  // end method displayResults

public void clear() {
itemCost = 0;
leaseCost = 0;
rate = 0;
time = 0;
salvageValue = 0;
maintCost = 0;

itemCostJField.setText( null );
leaseCostJField.setText( null );
rateJField.setText( null );
timeJField.setText( null );
salvageValueJField.setText( null );
maintCostJField.setText( null );

presentValueJLabel.setText( null );
differenceJLabel.setText( null );
rentalJLabel.setText( null );
maintContractJLabel.setText( null );
adviceJLabel.setText( null );

}  // end  method clear}
```