Comparison of Performance of Finite Element Codes in Different Programming Languages Converted From Legacy Finite Element Codes

Thiruchelvam Arudchelvam, Janaka Wijayakulasooriya, and S. Ratnajeevan H. Hoole

Abstract—The FORTRAN programming language was used in early days, to write programs for finite element based field computation. Much of those codes were developed in an ad hoc basis. Modern software developers face problems in understanding, modifying and utilizing those codes. As modern software engineers are more concerned on object oriented design, if those codes are converted into an object oriented language, those codes could be utilized in order to utilize the codes and redesign in an object oriented system. Those legacy codes are converted not only into an object oriented programming language such as Java but also into functional oriented languages such as C. Conversion of those legacy codes into modern languages gives many advantages. The purpose of this paper is to compare the performances of such legacy Finite element codes written in FORTRAN, the relevant converted C program and the relevant converted Java program. Finite element programs written in FORTRAN are converted into modern languages such as C and Java. Performances of those programs are compared based on the execution time. In addition to that, the memory sizes of the execution file of FORTRAN and C programs are compared.

Keywords— Finite Elements Method, Legacy Software, Performance Evaluation, Re-engineering.

I. LEGACY FINITE ELEMENT CODE

The development of code for finite elements-based field computation has been going on at a pace since 1970s. Today we have legacy code running into millions of lines, implemented without planning and not using proper state-of-the-art software design tools [1]. When a professor is requested to solve a problem, he/she developed codes with his research team. Then, the following researchers are also asked to use those codes to continue those researches. There were some threats in that approach. If the professor who knows the nook and hole of the project leaves the place, the whole project should thoroughly be studied by others in order to modify, expand and/or even to utilize those codes for some other purposes. Therefore, there is a need to convert those codes available in the FORTRAN language into some modern programming languages. Then it would be easier to the modern software developers and engineers to solve engineering problems. [2] describes the use of converting those legacy finite element codes into an object oriented model. Nanjundiah and Sinha, had enhanced FORTRAN codes into another later version of FORTRAN codes, [3]. Therefore, that work is just an enhancement of existing codes. Rather than enhancing legacy codes, they should be converted into an object oriented model so that it can ever be used in future. Therefore, it is suggested to apply reverse engineering in order to produce an object oriented model, [1].

II. RE-ENGINEERING FORTRAN

One of the most important aspects of reverse engineering would involve means of using legacy code written originally in FORTRAN since much of the legacy code is in FORTRAN. Now facilities are available to convert FORTRAN to C and Java [4], [5] and even from C to Java.

(a) Classes cannot directly be created from FORTRAN code using forward engineering. Therefore, to make it possible, FORTRAN code should be converted/translated first into Java code and it is thereafter those classes can be created, using reverse engineering.

(b) Earlier, FORTRAN programs were written based on a functional approach to programming rather than an object oriented approach. Therefore even if FORTRAN code is converted to Java code, it will not be in object oriented design.

III. CONVERSION OF FINITE ELEMENT CODE FROM FORTRAN TO C AND JAVA

Finite element program written in FORTRAN is converted into C and Java languages. The algorithm shown in Fig. 1 is used to compare the performances of programs in all three programming languages.

The number of mesh points is increased up to 1000 in the above algorithm. Therefore the same program is considered like several programs by changing the number of mesh points. Then the statistical analysis is carried out to compare the performances of those programs.

Further, the same algorithm is applied to three different finite element programs and the statistical analysis is carried out.
IV. STATISTICAL ANALYSIS

Graphs of execution times of FORTRAN, C and Java programs Vs. number of mesh points of programs 1, 2 and 3 are shown in Figs. 2, 3 and 4.

1. Number of mesh points in the program is set to 5.
2. The execution time of the FORTRAN program is calculated.
3. The memory size of the execution file of the FORTRAN program is also recorded.
4. FORTRAN program is converted into C.
5. FORTRAN program is converted into Java.
6. The execution time of the converted C program is calculated.
7. The execution time of the converted Java program is calculated.
8. Memory size of the execution file of the C program is also noted.
9. Number of mesh points is increased in FORTRAN program
10. Go to step 1.

IV. STATISTICAL ANALYSIS

HYPOTHESIS TEST FOR PROGRAM 1

Two-Sample T-Test and CI: FORTRAN, C

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>Mean</th>
<th>StDev</th>
<th>SE Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORTRAN</td>
<td>14</td>
<td>0.0820</td>
<td>0.0834</td>
<td>0.022</td>
</tr>
<tr>
<td>C</td>
<td>14</td>
<td>0.0766</td>
<td>0.0779</td>
<td>0.021</td>
</tr>
</tbody>
</table>

where N – number of programs; Mean – mean value; StDev – standard deviation; SE Mean - standard error of the mean. Difference = μ(FORTRAN) - μ(C)

Estimate for difference: 0.005429

95% CI for difference: (-0.057366, 0.068223)

Hypothesis Test:

H₀ – Execution time of FORTRAN = Execution Time of C
H₁ – Execution time of FORTRAN < Execution Time of C

T-Test of difference = 0 (vs not =):

T-Value = 0.18; P-Value = 0.860; DF = 25;-----------(1)

In this Two Sample T-Test, it was checked whether the performances of FORTRAN and C programs are equal or not. From (1), since P-Value is greater than 0.05, it can be concluded that the performances both FORTRAN and C programs are equal.

Two-Sample T-Test and CI: FORTRAN, Java

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>Mean</th>
<th>StDev</th>
<th>SE Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORTRAN</td>
<td>14</td>
<td>0.0820</td>
<td>0.0834</td>
<td>0.022</td>
</tr>
<tr>
<td>Java</td>
<td>14</td>
<td>0.297</td>
<td>0.159</td>
<td>0.043</td>
</tr>
</tbody>
</table>

where N – number of programs, Mean – mean value, StDev – standard deviation. SE Mean - standard error of the mean

Difference = μ(FORTRAN) - μ(Java)

Estimate for difference: -0.214500

95% CI for difference: (-0.315122, -0.113878)

Hypothesis test

H₀ – Execution Time of FORTRAN = Execution time Java
H₁ – Execution Time of FORTRAN < Execution time Java

T-Test of difference = 0 (vs <):

T-Value = -4.46; P-Value = 0.000; DF = 19;----------(2)

From (2), since P-Value is less than 0.05, it can be told that the execution time of FORTRAN is significantly less than the execution time of converted Java program. Therefore, it can be concluded that the performance of FORTRAN program is better than the performance of converted Java program.

Two-Sample T-Test and CI: C, Java

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>Mean</th>
<th>StDev</th>
<th>SE Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>14</td>
<td>0.0766</td>
<td>0.0779</td>
<td>0.021</td>
</tr>
<tr>
<td>Java</td>
<td>14</td>
<td>0.297</td>
<td>0.159</td>
<td>0.043</td>
</tr>
</tbody>
</table>

where N – number of programs, Mean – mean value, StDev – standard deviation. SE Mean - standard error of the mean

Difference = μ(C) – μ(Java)

Estimate for difference: -0.219929

95% CI for difference: (-0.319527, -0.120330)

Hypothesis test

H₀ – Execution Time of C = Execution time Java
H₁ – Execution Time of C < Execution time Java

T-Test of difference = 0 (vs <):

T-Value = -4.64; P-Value = 0.000; DF = 18;----------(3)

From (3), since P-Value is less than 0.05 it can be told that the execution time of C program is significantly less than the execution time of converted Java program.
HYPOTHESIS TEST FOR PROGRAM 2

Two-Sample T-Test and CI: FORTRAN, C_Prog

N  Mean  StDev  SE Mean
FORTRAN  15  0.385  0.372  0.096
    C_  15  0.379  0.370  0.096

Hypothesis Test:
H₀ – Execution time of FORTRAN = Execution Time of C
H₁ – Execution time of FORTRAN <> Execution Time of C
Difference = μ₁ (FORTRAN) - μ₂ (C_Prog)
Estimate for difference:  0.006467
95% CI for difference:  (-0.271580, 0.284513)
T-Test of difference = 0 (vs not =):
T-Value = 0.05;  P-Value = 0.962;  DF = 27 -------------(4)

In this Two Sample T-Test, it was checked whether the performances of FORTRAN and C programs are equal or not. From (4), since P-Value is greater than 0.05, it can be concluded that the performances both programs are statistically equal.

Two-Sample T-Test and CI: FORTRAN, Java

N  Mean  StDev  SE Mean
FORTRAN  15  0.385  0.372  0.096
Java      15  0.626  0.461  0.12

Hypothesis Test 1:
H₀ – Execution time of FORTRAN = Execution Time of Java
H₁ – Execution time of FORTRAN <> Execution Time of Java

Difference = μ₁ (FORTRAN) - μ₂ (Java)
Estimate for difference:  -0.240933
95% CI for difference:  (-0.555517, 0.073650)
T-Test of difference = 0 (vs not =):
T-Value = -1.57;  P-Value = 0.128;  DF = 26  =-------------(5)

From (5), since P-Value is greater than 0.05, it can be concluded that the execution time of FORTRAN is equal to the execution time of converted Java.

Two-Sample T-Test and CI: C_Prog, Java

N  Mean  StDev  SE Mean
C_Prog  15  0.379  0.370  0.096
Java      15  0.626  0.461  0.12

Difference = μ₁ (C_Prog) – μ₂ (Java)
Estimate for difference:  -0.247400
95% CI for difference:  (-0.561160, 0.066360)
T-Test of difference = 0 (vs not =):
T-Value = -1.62;  P-Value = 0.117;  DF = 26  =-------------(6)

From (6), since P-Value is greater than 0.05, it can be told that the execution time of converted C program is equal to the execution time of converted Java.
In this Two Sample T-Test, it was checked whether the performances of FORTRAN and C programs are equal or not. From (7), since P-Value is greater than 0.05, it can be concluded that the performances both programs are equal.

Two-Sample T-Test and CI: FORTRAN, Java

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>Mean</th>
<th>StDev</th>
<th>SE Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORTRAN</td>
<td>14</td>
<td>0.872</td>
<td>0.782</td>
<td>0.21</td>
</tr>
<tr>
<td>Java</td>
<td>14</td>
<td>1.41</td>
<td>1.08</td>
<td>0.29</td>
</tr>
</tbody>
</table>

Hypothesis test 1

H₀ – Execution Time of FORTRAN = Execution time of Java
H₁ – Execution Time of FORTRAN <> Execution time of Java

Difference = μ( FORTRAN ) – μ( Java )
Estimate for difference: -0.539929
95% CI for difference: (-1.277229, 0.197372)
T-Test of difference = 0 (vs not =):
  T-Value = -1.51; P-Value = 0.143; DF = 23 -------------(8)

As in the previous case, in this Two Sample T-Test, it was checked whether the performances of FORTRAN and converted Java programs are equal or not. From (8), since P-Value is greater than 0.05, it can be concluded that the performances both programs are equal.

Two-Sample T-Test and CI: FORTRAN, Java

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>Mean</th>
<th>StDev</th>
<th>SE Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORTRAN</td>
<td>14</td>
<td>0.872</td>
<td>0.782</td>
<td>0.21</td>
</tr>
<tr>
<td>Java</td>
<td>14</td>
<td>1.41</td>
<td>1.08</td>
<td>0.29</td>
</tr>
</tbody>
</table>

Hypothesis test

H₀ – Execution Time of FORTRAN = Execution time of Java
H₁ – Execution Time of FORTRAN <> Execution time of Java

Difference = μ( FORTRAN ) – μ( Java )
Estimate for difference: -0.539929
95% CI for difference: (-1.277229, 0.197372)
T-Test of difference = 0 (vs not =):
  T-Value = -1.51; P-Value = 0.143; DF = 23 ---(8)

From (10), since P-Value is greater than 0.05, it can be concluded that the performances both programs are equal.

The following results are obtained when the average values of all the above are taken and the same statistical tests are done.

1. Execution time of FORTRAN program and the execution time of converted C Program are equal.
2. Execution time of FORTRAN program and the execution time of converted Java Program are equal.
3. Execution time of converted C program and the execution time of converted Java Program are equal.

The same types of tests are done for the memory sizes of the execution files of FORTRAN program and the converted C program. From those tests the following result is obtained:

“The memory size of the execution file of FORTRAN is less than that of converted C program.”

Java programs are having only class files and not execution files. Therefore, memory size of the execution file of Java programs is not considered for this test.

From the above statistical analysis, when memory sizes of the execution files of FORTRAN program and the converted C program are considered, the performance of the FORTRAN program is better. But, when execution time is considered the performances of programs in all three programming languages are same. Further, converting FORTRAN programs into C and Java gives many advantages, [1], [2]. Therefore, converting FORTRAN programs into C and Java does not decrease the performance of programs.

V. CONCLUSION

This paper has statistically analysed the performances of FORTRAN and relevant converted C and Java programs. The memory size of the execution files and the execution times are considered as two factors of the performance indicators. Based on these two factors, the performances of the programs are compared. Accordingly, we tested hypothesis that the execution times are the same. Our finding is that, on average, the performances of the finite element FORTRAN program, the relevant converted C program and the relevant converted Java program are statistically equal. Further, according to our hypothesis tests, the memory sizes of the execution files of the
FORTRAN program are statistically less than that of the relevant C program. The memory test was not relevant to the Java program.

ACKNOWLEDGMENT
Mr. Thiruchelvam Arudchelvam thanks his employer, Wayamba University of Sri Lanka, for the study leave to carry out this research; University of Peradeniya Sri Lanka where he is a research student; and Ms. W.M.P.M. Wickramasnghe and Ms P.A.D.A.N.Appuhamy for the support rendered for statistical analysis.

REFERENCES


