Stability and performance study of models in 32 and 64 bits

BAE SADM and DRDC MATLAB/Simulink models

Martin Camirand, ing jr.
LTI informatique et genie

Prepared By:
LTI informatique et génie
1305 Boul. Lebourgneuf, Bureau 130
Québec, Qc, Canada, G2K 2E4
Contract Project Manager: Martin Soucy
PWGSC Contract Number: W7701-155932
Technical Authority: Éric Dumais and Alfred Jeffrey, Weapons Systems Section, DRDC – Valcartier Research Centre

Contract Report
DRDC-RDDC-2016-C220
January 2016

ADMINISTRATIVE STATEMENTS

This document was reviewed for Controlled Goods by DRDC using the *Guide to Canada's Export Controls*.

The authors would also like to acknowledge the support of the DRDC Navy / Above Water Warfare (AWW) / Direct Client Support project and the Canadian Forces Maritime Warfare Centre (CFMWC), whose financial support facilitated the development of processes and tools to analyse the stability and the performance of DRDC weapons models in MATLAB/Simulink and in the BAE Ship Air Defence Model (SADM) in 32 and 64 bits. Finally, the authors would like to acknowledge Mr. Alfred Jeffrey (Contract Scientific Authority), and Mr. Eric Dumais for their guidance and support. This work was performed between October 2015 and February 2016 under PWSGC contract W7701-155932/001/QCL Firm Task.

Abstract

The development of 64-bit applications has become more important in the last couple of years and it even seems to have become the standard. The migration to a 64-bit version of MATLAB for the DRDC Valcartier Weapons Systems section was not really an option since Mathworks announced that the R2015b version of MATLAB will be the last release of the 32-bit version of the software for Windows. This means that the Weapons Systems Library (WSL), which is MATLAB/Simulink based, would no longer be supported for future versions. In addition, various softwares which are MATLAB based will also need to be updated to 64-bit, which is mainly due to software components interoperability with the 64-bit versions of MATLAB that requires that the other software also be 64-bit.

As part of the *W7701-155932 R&D Weapon System* project, the mandate regarding the system migration was mainly to make sure that such a migration would not impact on model stability, compiling behavior, and scenarios results within the DRDC MATLAB/Simulink library as well as in the BAE SADM simulation software. The current report will study the effects of the aforementioned software migration on scenarios simulation results from both SADM and WSL and their use of DRDC library models, by investigating the parameters initialization, run time and compilation time. Computing performances assessment will be performed using statistical tests in order to compare 32 and 64-bit tests and to check for substantial variation on samples.

The results of those tests do not show a significant variation within the SADM simulation results, and also demonstrate that the 64-bit migration is not significantly altering stability or computing performance of MATLAB and SADM for scenarios below 500 Monte Carlo (MC) trials.

Significance for Defence and Security

The current performance and stability analyses are of great interest for the Canadian Forces (CF) as they exhaustively assess the actual reliability of software used by the DRDC Valcartier Weapons Systems section to develop and analyse models and scenarios of actual weapons or weapons systems in order to study engagements outcomes.

The design and implementation of such numerical models allow one to simulate the behaviour of various weapons in different engagement scenarios, such as the air-to-ground or surface-to-air engagement model, is of great interest for the CF. Indeed, these models allow for a rapid assessment of weapon performance in various situations or can be used to support future weapon definition, procurement and operation.

Because of the various usages of numerical models by the CF, the necessity to make sure that a software migration to a 64-bit set-up is providing reliable data with a time effective computing process was critical.

This page intentionally left blank.

Résumé

Le développement d'applications 64-bit est devenu de plus en plus important dans les dernières années et il semble que cela soit de plus en plus amené à devenir le standard de l'industrie. La migration vers une version 64-bit de MATLAB pour la section Systèmes d'armes de RDDC Valcartier n'était pas vraiment un choix car Mathworks a annoncé que la dernière version 32-bit Windows de MATLAB serait R2015b. Cela signifie que les modèles de la librairie de la section Systèmes d'armes (LSA) principalement définie sur une base MATLAB / Simulink ne seraient plus soutenus pour les versions à venir. Outre le logiciel de MathWorks, certains autres logiciels implémentant MATLAB devront également subir une migration 64-bit car l'interopérabilité avec les versions 64-bit de MATLAB requiert que les logiciels soit également sous une base 64-bit.

Dans le cadre du projet *W7701-155932 R&D de systèmes d'armes*, le mandat qui concerne la migration logicielle a impliqué principalement de faire une étude d'impact pour une telle migration et de s'assurer qu'elle n'affecte pas la stabilité des modèles, le comportement de compilation et les performances de calcul que ce soit avec MATLAB / Simulink ou SADM. Le présent rapport étudie les effets d'une migration du logiciel sur les résultats de simulation des scénarios dans SADM ainsi que ceux dans la librairie de la section Systèmes d'armes en analysant les performances d'initialisation de paramètres de modèle, d'exécution de scenario et de compilation de fichiers. La comparaison de ces performances de calcul se fera à l'aide de tests statistiques pour comparer les résultats obtenus sous 32 et 64 bits.

Les résultats de ces tests démontrent une certaine variation dans les résultats de simulation SADM, qui sera estimée comme négligeable pour le moment, mais tout de même suffisamment significative pour soulever un questionnement. Il est également démontré que la migration 64-bit n'altère pas de manière significative la stabilité ou la performance de MATLAB ni de SADM pour les scénarios en dessous de 500 essais MC.

Importance pour la défense et la sécurité

Les analyses de performance et de stabilité dont il est question dans le présent rapport sont d'un grand intérêt pour les Forces canadiennes (FC), car ces logiciels sont utilisés par la section Systèmes d'Armes de RDDC Valcartier pour développer et analyser des modèles et des scénarios d'armes ou d'engagement.

La conception et la mise en œuvre de ces modèles numériques, capables de simuler le comportement de diverses armes dans différents scénarios d'engagement, tels que l'air-sol, surface-air, est d'un grand intérêt pour les FC. En effet, ces modèles permettent une évaluation rapide de la performance d'arme dans diverses situations ou peuvent également être utilisés pour définir la viabilité d'une arme ou encore la meilleure manière de l'utiliser.

En raison de ces différentes utilisations des modèles numériques par les FC, il est facile de comprendre la nécessité de s'assurer que la migration d'un logiciel à un format 64-bit fournisse des données toujours fiables, et ce, avec un processus de calcul efficace.

This page intentionally left blank.

Table of contents

List of figures

List of tables

This page intentionally left blank.

Acknowledgements

The authors would like to acknowledge the support of the DRDC Navy / Above Water Warfare (AWW) / Direct Client Support project and the Canadian Forces Maritime Warfare Centre (CFMWC), whose financial support facilitated the development of processes and tools to analyse the stability and the performance of DRDC weapons models in MATLAB/Simulink and in the BAE Ship Air Defence Model (SADM) in 32 and 64 bits. This work was performed between October 2015 and February 2016 under PWSGC contract W7701-155932/001/QCL Firm Task.

Finally, the authors would like to acknowledge Mr. Alfred Jeffrey (Contract Scientific Authority) and Mr. Eric Dumais (Subject Matter Expert) for their guidance and support.

This page intentionally left blank.

1 Introduction

The development of 64-bit applications has become more important in the last couple of years and it even seems to have become the standard. A 64-bit system can still operate a 32-bit software. However, the migration to a 64-bit operated software for the DRDC was not really an option since Mathworks announced that the R2015b version of MATLAB will be the last release of the 32-bit version of the software for Windows. That means that the DRDC Weapons Systems Library (WSL), which is MATLAB/Simulink based, would no longer be supported for future versions. In addition, various softwares which are MATLAB based will also need to be updated to 64-bit, which is mainly due to software components interoperability with the 64-bit versions of MATLAB that requires that the other software also be 64-bit.

As a part of the *W7701-155932 R&D Weapon System* project, the mandate regarding the system migration was mainly to make sure that such a migration would not impact on model stability, compiling behavior, and scenario results within the DRDC MATLAB/Simulink library as well as in the BAE SADM simulation software. Since this specific software usually works with large data sets and execution time consuming simulations (batch runs), it was also deemed highly relevant to evaluate the performance variation linked to a 64-bit migration.

For countless infrastructures, the gain in performance of moving to 64-bit environments is unclear. One of the most obvious advantages of the 64-bit environment is that it removes the 4GB of memory limit that exists on 32-bit machines which may sound interesting at first glance. However, depending on the application/environment, the impact can be anywhere from null to significant. It can be easily illustrated by considering that the fastest 8-cylinder car can be a real asset when on a long stretch of open road but that it loses its edge in bumper-to-bumper city traffic. Not knowing the exact program architecture and data processing sequencing, the only way to evaluate accurately the performance variation was to run various tests on both versions and compare results.

The current report will present the results of those comparisons: on SADM scenario simulation results, on performance analysis done on the DRDC WSL, on parameters initialization, on run time and finally on compilation time. Performance analysis of the SADM scenario will also be presented for various number of Monte Carlo (MC) trials. While Section 2 presents the direct SADM simulations results, Section 3 focuses on the study of performance oriented tests to compare 32 and 64 bits computing performances when using MATLAB/Simulink and SADM. Finally, Section 4 summarizes all the information gathered and conclusions drawn from these tests.

This page intentionally left blank.

2 SADM simulation results comparison

The idea of doing a simulation results comparison prior to a performance analysis is simply to ensure that during the 64-bit migration, that the software shows a recurrent behavior and the interoperability between SADM and MATLAB is still viable. Finally it is used to validate that a 64-bit set-up produces results that are consistent with those obtained on a 32-bit set-up.

2.1 Methodology

The SADM version 5.3.0.1 in both 32-bit and 64-bit was used to run the preliminary tests.

The methodology of the tests consists in running a given DRDC designed scenario *(ActiveMissile.scn)* that is calling an external DRDC missile model *(ActiveMissile.wpn)*. The scenario should be tested in 32 and 64 bits versions with a sufficient number of trials to have reliable sets of data. The obtained results are then compared.

The different criteria on which results are going to be compared are the probability of survival (Ps) for a given scenario, the average event timing of the simulation and some of the parameters specific to the external missile such as the fly-out pattern and the weapon velocity profile. Comparing those data should provide roughly an accurate confirmation that the software migration has not caused any instability within the simulation process and produces simulation results that are similar enough to be considered equivalent.

2.2 Results comparison for 32 vs 64 bits

The DRDC *ActiveMisisle* Model was used as the test case. The model was executed and specific model outputs were compared to test the migration of the given model from 32 to 64 bit.

2.2.1 Probability of survival (Ps)

The probability of survival (Ps) was determined to be the first criteria of comparison since it is ultimately the final outcome result of a scenario and it should represent a rather good indicator of the simulation stability. Table 1 shows those results for the Ps in the *ActiveMissile* scenario

Table 1 : *Results for the probability of survival (Ps) in the ActiveMissile scenario.*

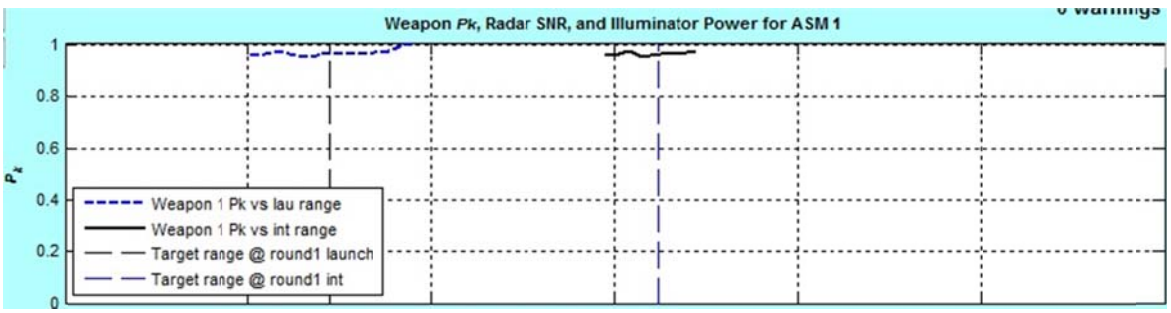
Average probability of survival (PS)		
Nb of MC	**32-bit**	**64-bit**
100	96%	94%
1000	96.30%	94.60%

As shown in Table 1, the first set of tests done with 100 MC trials gave a 2% difference between the 32 and 64 bits version. 2% is not a substantial difference but it is noticeable. For verification purposes, it was decided to run a more substantial number of MC trials to make sure that the variation observed was recurrent. The 1000 MC trials tests are close but still reveal 1.7% of difference. For now, since the results can be considered to be close enough, this difference would be assumed to be negligible, but the study of the following measure, the probability of kill, might be useful to confirm or deny this assumption.

2.2.2 Weapon probability of kill (Pk)

The probability of survival for a given scenario is directly linked to the Pk table and Pk evaluation for a given weapon. A variation in the Pk values could explain the aforementioned variations observed on the probability of survival (Ps). Figure 1 shows the SADM Weapon Pk 32 vs 64 bits comparison for the *ActiveMissile* scenario.

32-bit



64-bit

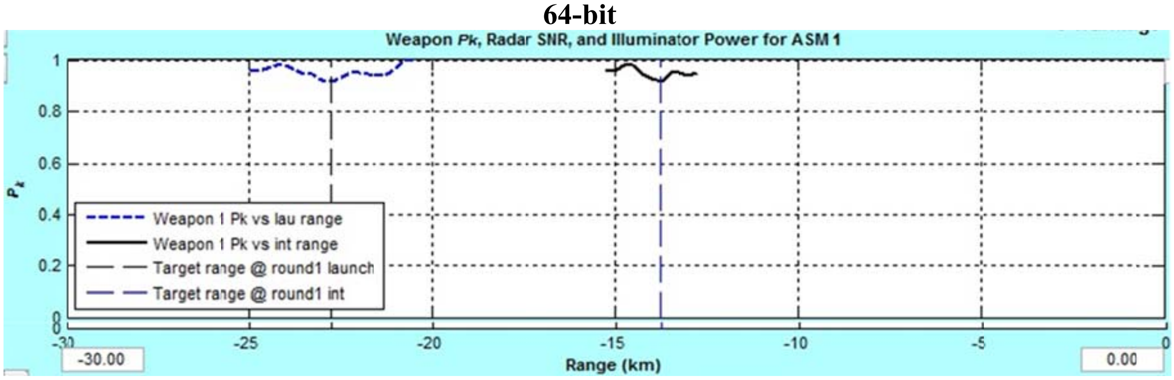


Figure 1 : *SADM Weapon Pk 32 vs 64 bits comparison for the ActiveMissile scenario for a 1000 MC trial.*

The Pk graphical information presented in Figure 1 shows a small variation between the 32 and the 64 bits for a 1000 MC run. This small variation might explain the variations observed on the Ps. Further investigations to explain this variation reveal that the .wpn and .scn files used by the 32 and 64 bits simulation are identical. Therefore, this variation on the Pk and on the Ps cannot be easily explained since every user controlled data is identical and the computing data and processing code cannot be monitored easily within the SADM software. However, with only a 2% variation, at this moment, the difference will be assumed to be not significant enough to conclude that the simulation analysis results are too different. The analysis will be pushed on further to evaluate other simulation parameters in order to check for other irregularities.

2.2.3 Scenario Events

The next step in the results investigation compares the main events parameters in the simulation, such as the range and time of detection, range and time to track, time to engage the incoming treat, and time to kill. In that vein, Figure 2 shows the SADM resource Timeline 32 vs 64 bits comparison for the *ActiveMissile* scenario.

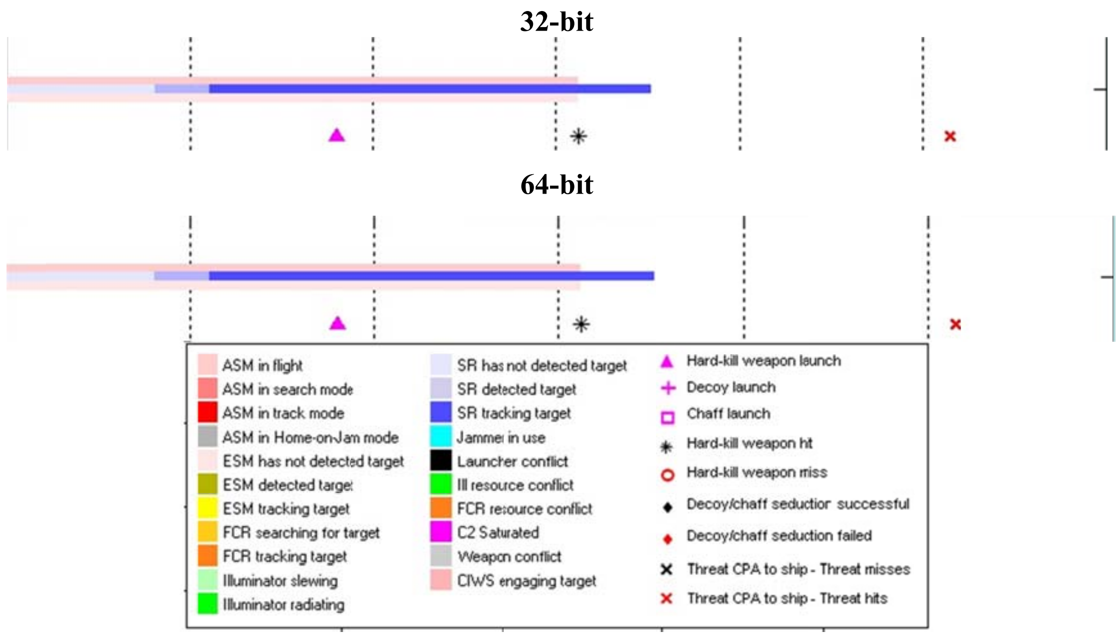


Figure 2 : *SADM resource Timeline 32 vs 64 bits comparison for the ActiveMissile scenario.*

As illustrated in Figure 2, the resource timeline provided by SADM seems to show a very consistent timing for both scenarios. Table 2 shows average results for various events in the *ActiveMissile* scenario.

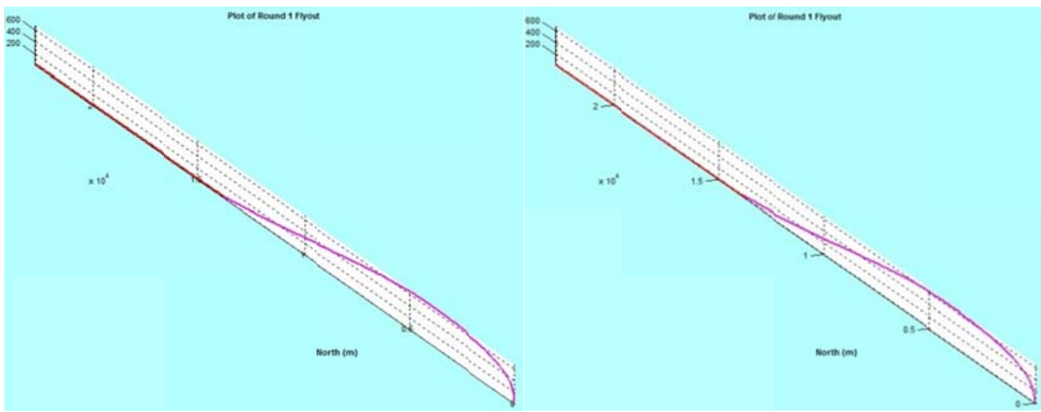
Table 2 : *Average results for various events in the ActiveMissile scenario.*

Average scenario events values		
Parameters	**32-bit**	**64-bit**
Avg. detect range (km)	29.496	29.496
Avg. time to detect (s)	8.1	8.1
Avg. track range (km)	28.474	28.474
Avg. Time to track (s)	9.6	9.6
Avg. Time to engage (s)	17.6	17.6
Avg. Time to kill (s)	31	31

As shown in Table 2, one observes that the event time and range averages are equivalent for both the 32 and 64-bit.

2.2.4 Fly-out trajectory

As scenario events timing have been confirmed to be identical, various other simulation parameters were observed and compared to ensure that the simulation produces comparable results. One of the parameters that have previously proven to be problematic[1] is the fly-out profile of an external missile. Accordingly, Figure 3 shows SADM target and interceptor fly-out trajectories for both the 32 vs 64 bits models.



32-bit *64-bit*

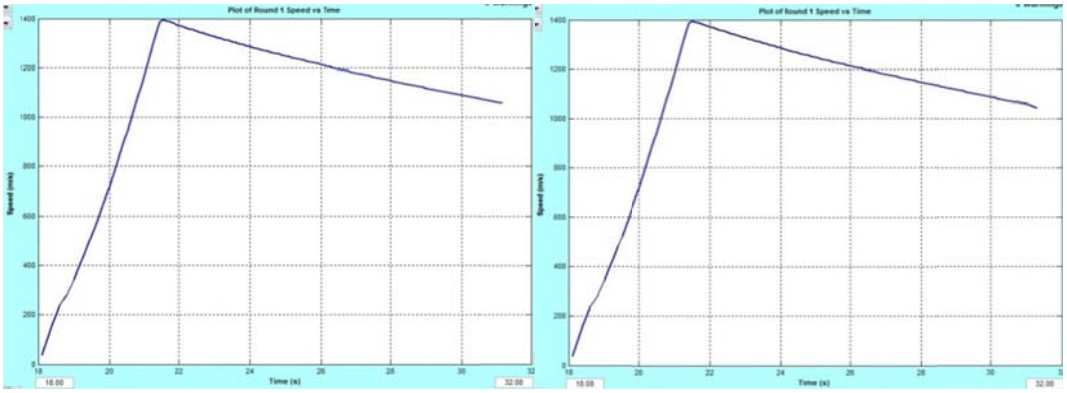
Figure 3 : *SADM target (red) and interceptor (pink) fly-out trajectory 32 vs 64 bits comparison .*

As shown in Figure 3, both fly-out pattern (target and interceptor) prove to be rigorously identical, thus demonstrating that the environment physics applied on the models is being conserved.

2.2.5 Weapon Speed vs Time

The previous trajectories comparison showed a level of similarity that should confirm the physic laws of the environment as well as the physic laws relative to the missile itself. To have a second level of confirmation, the velocity profile for the weapon should also provide information that proves that the external missile propulsion system is correctly considered. Thus, Figure 4 shows the SADM weapon speed vs time graph for both the 32 vs 64 bits models.

[1] Problem on the fly-out pattern has mainly been observed on SADM version change.



32-bit *64-bit*

Figure 4 : *SADM weapon speed vs time graph 32 vs 64 bits comparison.*

As shown in Figure 4, the speed profiles show little to no difference, confirming the conservation on physics laws applied on the environment, propulsion system and missile aerodynamic.

2.3 Conclusion on simulation results viability

Overall, the general behavior of the simulation does not seem affected by the 64-bit migration. All of the main events of interception are occurring in the same exact order, same range and timing. The general physics laws applied within the simulation seem to have been conserved.

The only concern is the Ps and Pk, variation observed that does not seem consistent with the level of accuracy showed on other parameters. Nevertheless, with only a 2% variation with every other aspect of the scenario identical, at this moment, the difference observed will be assumed to be not substantial enough to conclude that the simulation analysis results are too different to be compared. However, it might be a good idea to investigate the issue further with larger sets of data or in conjunction with BAE in the interest of future simulation that might require a very rigorous stability in scenarios results.

This page intentionally left blank.

3 Performance based comparison

It has been demonstrated that the scenarios results were not altered by the system migration from 32 to 64 bit. Now, a more extensive performance based analysis will be performed. The idea of doing a performance monitoring by comparing execution time of various MATLAB and SADM processes is simply to ensure that during the 64-bit migration, software show a recurrent behavior, or even a gain, on their computing time. It is also a way to check that the interoperability between SADM and MATLAB is still viable and as efficient as it was previously observed. Finally it will be used to validate that the 64-bit set-up produces results that are consistent with those obtained on a 32-bit set-up as well as to quantify the general gain in computing performances.

3.1 General computer and server specs

Execution time for a given program will obviously be influenced by it physical set-up (32 vs. 64-bit), and by the specifications of the hardware used for execution. That means that the results obtained might differ from a computer to another. For reference, for further performance analysis and unless otherwise specified, the current results discussed in this report were generated using a computer with the following specifications:

- Windows 7 Enterprise, 64 bits Operating system (SP 1)
- Intel® Core™ i7 CPU 980 @ 3.33GHz 3.33GHz
- 24 GB of RAM

However, it important to mention that the tests done on the model compiling time could not be performed on the same hardware due to Simulink Coder licence management. Those tests were done on a virtual machine (VM) with these specifications:

- Windows Server 2008 R2 Standard (SP 1)
- Common VM processor 2.60 GHz (2 processors)
- 32.0 GB of RAM

In addition to the previously mentioned factors, other processes, activities and software on the test computer might be susceptible to have an impact on performance results as well. For that reason, it is worth noting that tests were done by ensuring minimal test computer external activity. Finally, trying to recreate the exact same simulation conditions might prove to be difficult but it is believed that the multiple batch runs should minimise the irregularities. Furthermore, even if running time changes for a similar set-up, the overall difference between the 32 and 64 bits should be similar or at least in proportion.

3.2 MATLAB/Simulink

Running performance analysis on the MATLAB components is essential since the WSL is MATLAB/Simulink based. This software is also used by the BAE simulation and analysis software SADM, so before monitoring a dual software interaction, an evaluation was performed initially on the MATLAB component.

3.2.1 Version and methodology

All the tests done using MATLAB an SADM were done with these MATLAB versions:

- 32-bit R2013b (8.2.0.701)
- 64-bit R2013b (8.2.0.701)

The methodology used to evaluate the performance, of each given test, was to create a simple program to run batches of initialisations, simulations and compilations and to monitor them using the "tic" and "toc" MATLAB functions prior to stock the results in a matrix from witch data can be later extracted to run statistical analysis.

The number of tests performed for each step had to be large enough to avoid data contamination with irregularities and also to make sure that it can be extrapolated to a Gaussian/normal distribution (for the sake of simplifying further performance comparison). To ensure that the variations in the simulation time can mainly be attributed to randomness, each and every batch of compiled data would have to be tested for normality. Indeed, a level of randomness was observed within the performance tests results. However, they can also be influenced by various other parameters. The normality would be necessary since the comparison analysis to be was based on the assumption of a Gaussian distribution. The normality test is a way to ensure that randomness is the main variation factor within the tests and that no other external factors may influence the results, as previously discussed. The normality assumption should be taken seriously, for when this assumption does not hold, it is impossible to draw accurate and reliable conclusions about the mathematical reality and the data correlation.

The analysis of the data can be simplified by ensuring that the samples are large enough (> 30 or 40). With a big enough sample, a violation of the normality assumption should not cause any major problems. If we have samples consisting of 100 observations, the distribution of the data can be ignored [1]. Indeed, according to the second fundamental theorem of statistics, namely the central limit theorem:

- *If "Sn" is the sum of "n" mutually independent random variables, then the distribution function of "Sn" is well-approximated by a certain type of continuous function known as a normal density function [2]*

This means that if the samples of data are approximately normal, then the sampling distribution will be normal as well. Although true normality is generally considered to be a myth, a certain degree of normality could be obtained and considered close enough to work with a normality assumption. Normality could be visually identified by using normal plots or by significance tests which mainly consist of comparing the sample distribution with one that follows the normal law.

For the purpose of the report, the normality of the gathered data are going to be visually analyzed (to look for the typical normal distribution bell curve) and mathematically confirmed by running a χ2 test. The χ2 goodness-of-fit test might be a little less known than other normality tests such as the Kolmogorov-Smirnov, the Anderson-Darling or the Shapiro-Wilk tests, however it was retained as our significance test because it is less complicated, robust and easier to implement on large sets of data than other test types.

The χ2 test is based on the difference between the observed and the expected values for each set of data. This is done by calculating the χ2:

$$\chi^2 = \sum_i \frac{(O_i - E_i)^2}{E_i} \quad (1)$$

where "O_i" is the observed data, and "E_i" is the expected data. This expected value will depend on the null hypothesis which can be defined as:

$$H_0: O_i = E_i \quad (2)$$

To confirm the null hypothesis, the χ2 values obtained need to be compared to critical values obtained with a χ2 table and determined if either equal or less than the critical value for a level of confidence α. Table 3 shows χ^2 critical values for a lower tail probability α with "df" degrees of freedom.

Table 3 : *χ^2critical values for a lower tail probability α with "df" degrees of freedom.[3]*

df	$\chi^2_{.995}$	$\chi^2_{.990}$	$\chi^2_{.975}$	$\chi^2_{.950}$	$\chi^2_{.900}$	$\chi^2_{.100}$	$\chi^2_{.050}$	$\chi^2_{.025}$	$\chi^2_{.010}$	$\chi^2_{.005}$
1	0.000	0.000	0.001	0.004	0.016	2.706	3.841	5.024	6.635	7.879
2	0.010	0.020	0.051	0.103	0.211	4.605	5.991	7.378	9.210	10.597
3	0.072	0.115	0.216	0.352	0.584	6.251	7.815	9.348	11.345	12.838
4	0.207	0.297	0.484	0.711	1.064	7.779	9.488	11.143	13.277	14.860
5	0.412	0.554	0.831	1.145	1.610	9.236	11.070	12.833	15.086	16.750
100	67.328	70.065	74.222	77.929	82.358	118.498	124.342	129.561	135.807	140.169

As shown in Table 3, the degree-of-freedom of a typical χ2 table does not reach beyond 100. On the other hand, the degree-of-freedom of our data would be equal to ∞ since they could be theoretically any value. However, to work with a degree of freedom of 100, would be considered to be reasonably accurate because the chi squared distribution is asymptotically normal as the number of degrees of freedom becomes infinite. For the χ^2 test, it is scientifically accepted to consider that a 95% of confidence ($\chi^2_{.050}$) is sufficient to confirm the null hypothesis.

To test the normality of a set of data would simply mean to test it with the average as an expected value. The same test could be used to evaluate if two sets of normally distributed data can be considered as equivalent. For that, the null hypothesis should use the expected data to be either the other sample iteration value or the other mean value.

3.2.2 Results

Since detailed calculations on such a large set of data would mean a lot of recurrent and not so useful information, only the mainstream data will be presented. Methodology presented at the beginning of this chapter for test validation has been covered, and unless stated otherwise this approach will be used to validate the results. Therefore, this section will present the information mainly with graphical results, key statistical values and χ^2 test results without detailed calculations.

3.2.2.1 Model parameters initialisation *(GenerateActiveMissileParameters)*

The DRDC library uses models that often require some basic parameter initialisation to allow for model operation. In the case of the *ActiveMissile* model, initialisation is performed through the *GenerateActiveMissileParameters* command. This was the first command on which the performance test was run. Figure 5 shows the set of data compiled for the model's parameters initialisation time in 32-bit.

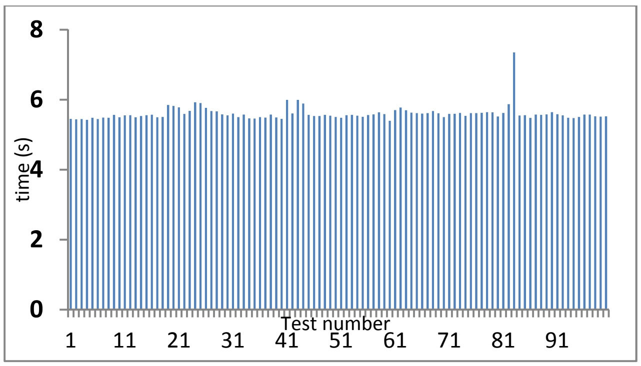


Figure 5 : *Set of data compiled for the model's parameters initialisation time in 32-bit.*

As shown in Figure 5, the regularity of the data is quite obvious except with one observed irregularity. Overall one observes a constant computing time.

Table **4** presents the main statistical information about this specific set of data.

Table 4 : *Main statistical values for the 100 run on the 32-bit parameters initialization.*

GenerateActiveMissileParameters 32-bit	
min	5.3986
q1	5.5066
median	5.5689
q3	5.6202
max	7.3511
mean	5.6078
sd	0.2122

As shown in Table 4, these values allow us to roughly validate that the quartiles are equally distributed, respecting the bell curve shape, where the difference between the median and the mean shows a slightly positive skewness. A look at the data distribution and the normal curve of the data should confirm those hypotheses and allow a first normality assumption. Figure 6 shows the model parameters initialisation time data distribution and normal curve in 32-bit.

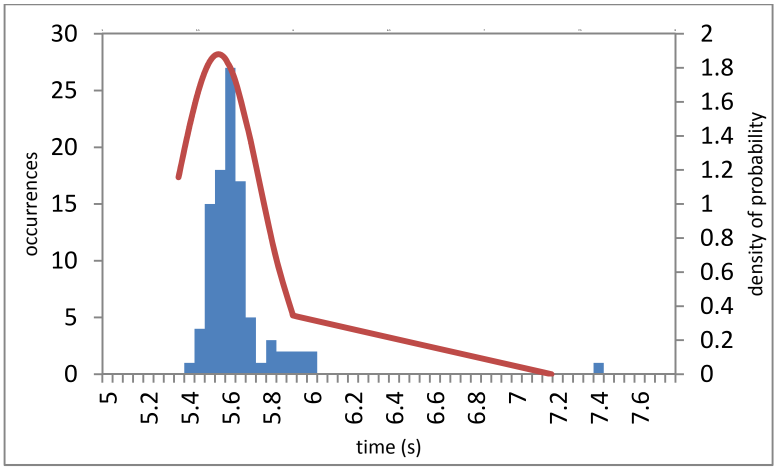


Figure 6 : *Model's parameters initialisation time, data distribution and normal curve in 32-bit.*

Figure 6 clearly shows a bell curve shaped graph responding to the first description mentioned, and representative of a normal distribution curve. The χ^2 test result should confirm the null assumption, as follows:

$$H_0: O_i = 5.60775 \tag{3}$$

$$\chi^2 = 0.8029 \tag{4}$$

The critical value for a level of confidence of 95% would be $\chi^2_{.050} = 124.342$, with the value obtained we then easily confirm the null hypothesis and accept that the distribution can be considered as a normal distribution. Similarly, Figure 7 presents the set of data compiled for the model's parameters initialisation time in 64-bit.

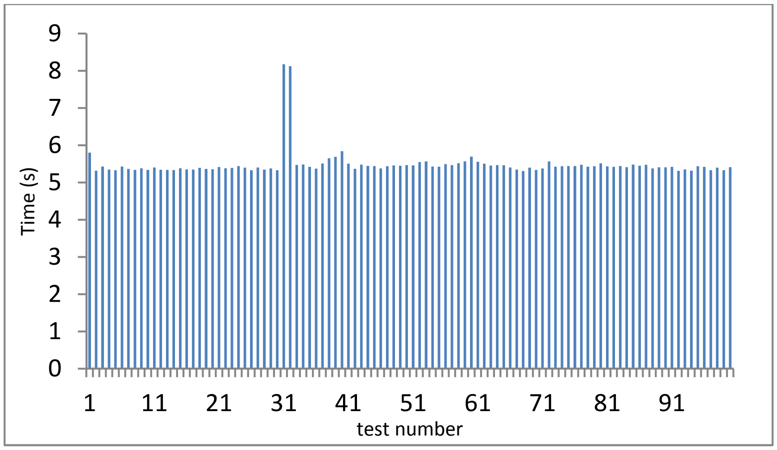


Figure 7 : *Set of data compiled for the model's parameters initialisation time 64-bit.*

As shown in Figure 7, as for the 32-bit data, the regularity in the 64-bit parameter initialisation times are quite obvious. Again one observes some irregularities, but not too concerning due to their low occurrences. Table 5 lists the main statistical values for the 100 runs on the 64-bit parameters initialisation.

Table 5 : *Main statistical values for the 100 run on the 64-bit parameters initialization.*

GenerateActiveMissileParameters 64-bit	
min	5.3067
q1	5.3738
median	5.4233
q3	5.4673
max	8.1746
mean	5.4870
sd	0.3919

Again, as shown in Table 5, the data is roughly symmetrical with a slightly positive skewness. Figure 6 shows the model parameters initialisation time data distribution and normal curve in 64-bit.

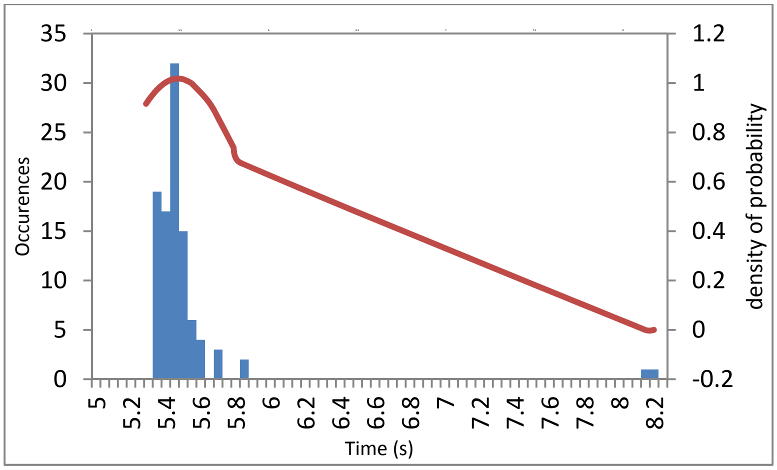


***Figure 8 :** Model's parameters initialisation time data distribution and normal curve in 64-bit.*

Figure 8 shows a normal curve that is close enough to a normal distribution curve, at least for the area over the main distribution. The normal graph deformation on the right side is mainly due to the observed two abnormalities, but their low occurrence should mean they are negligible, which should be confirmed by the χ^2 test result.

$$H_0 : O_i = 5.4870 \tag{5}$$

$$\chi^2 = 2.7988 \tag{6}$$

As previously mentioned, the critical value for a level of confidence of 95% would be $\chi^2_{.050} = 124.342$. With the value obtained we easily confirm the null hypothesis and accept that the distribution of this sample can be considered as a normal distribution.

Now that it has been demonstrated that both samples can be considered as Gaussian distributions, they can legitimately be compared to each other, which is illustrated in Figure 9.

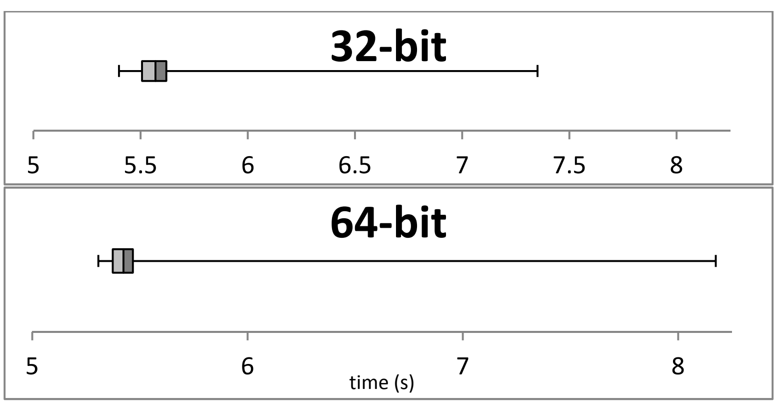


***Figure 9 :** Quartile distribution for parameters initialisation time test.*

The quartile graph of the two functions of Figure 9 shows a distribution of data that is quite similar, except for the maxima. However, as previously discussed, the observed maxima are more of an irregularity within the two sets of data and with their low occurrence they should not be considered as a determinant factor. The χ^2 test will be run on the data with two different null assumptions:

$$H_{01}: O_{i64} = O_{i32} \quad \text{and} \quad H_{02}: O_{i64} = Mean_{32} \tag{7}$$

$$\chi^2_{01} = 3.7775 \quad \text{and} \quad \chi^2_{02} = 2.9985 \tag{8}$$

Both χ^2 test result show values considerably smaller than the critical value of 124.342, meaning that both null assumptions are confirmed. This confirms that it is ok to assume that the two set of data can be considered as equivalent, showing no significant performance variation in MATLAB for the scenario parameters initialization.

3.2.2.2 Model run time *(AMdll.mdl)*

Once the model has been initialised, the next step would be to check for the model MATLAB run time. In that vein, Figure 10 shows the set of data compiled for the model's run time in 32-bit.

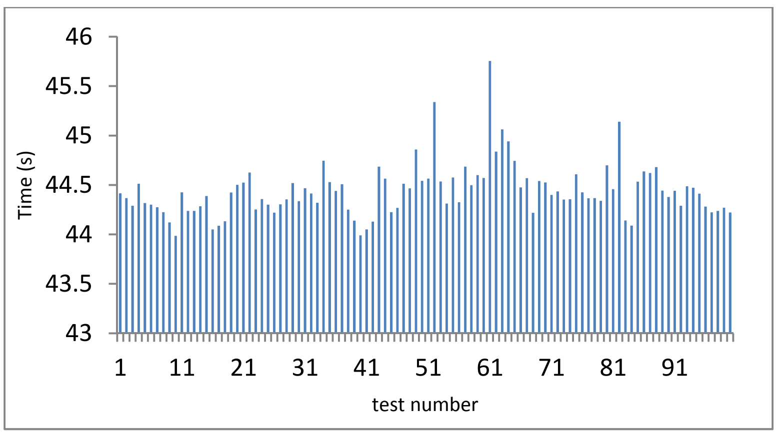


Figure 10 : *Set of data compiled for the model's run time in 32-bit.*

Figure 10 shows the regularity of the data, although somewhat less apparent than that observed on the previous test. Indeed, the current gathered data shows somewhat greater variation and some irregularities. However, the data seems to be somewhat equally distributed to allow one to construct an imaginary mean line. The variation might be explained by the fact that this computing sequence considers more factors susceptible to randomness. Table 6 lists the main statistical values for the 100 runs performed on the 32-bit *AMdl.mdl* run time.

Table 6 : *Main statistical values for the 100 runs on the 32-bit AMdl.mdl run time.*

AMdll.mdl run time 32-bit	
min	43.9853
q1	44.2800
median	44.4156
q3	44.5364
max	45.7547
mean	44.4399
sd	0.2656

As shown in Table 6, for this sample, statistical values are showing a roughly symmetrical distribution of the data with almost no skewness. This is giving a good indicator of a Gaussian distribution. Figure 11 shows the model run time data distribution and normal curve in 32-bit.

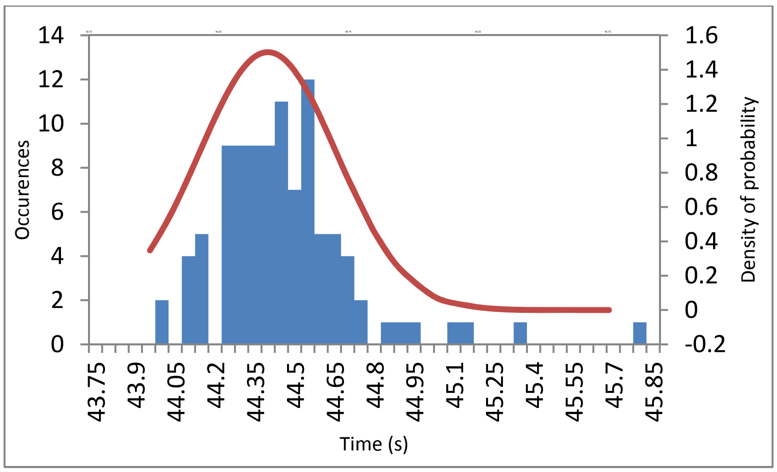


***Figure 11 :** Model's run time data distribution and normal curve in 32-bit.*

The data distribution and normal graph of Figure 11 show an almost perfect bell curve shaped graph typical for a normal distribution. The χ^2 test result confirms the null assumption, as follows:

$$H_0: O_i = 44.4399 \quad (9)$$

$$\chi^2 = 0.1587 \quad (10)$$

Again, the χ^2 test shows a value that is significantly lower than the critical value of $\chi^2_{.050} = 124.342$. This confirms that this set of data may be considered as a normal distribution. Figure 12 shows the set of data compiled for the model's run time of 64-bit.

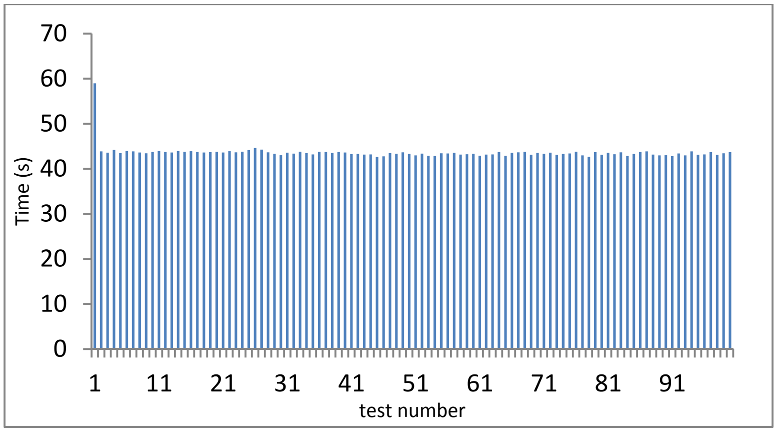


Figure 12 : *Set of data compiled for the model's run time in 64-bit.*

Figure 12 shows the high regularity of the run time with the 64-bit set-up, being almost constant with only one irregularity. Table 7 lists the Main statistical values for the 100 runs on the 64-bit *AMdl.mdl* run time.

Table 7 : *Main statistical values for the 100 runs on the 64-bit AMdl.mdl run time.*

AMdll.mdl run time 64-bit	
min	42.6356
q1	43.1811
median	43.4940
q3	43.7176
max	58.9992
mean	43.6169
sd	1.5887

Statistical values of Table 7 show a very symmetrical and concentred (irregularity apart) distribution of the data with a slightly positive skewness. Figure 13 illustrates the model run time data distribution and normal curve in 64-bit.

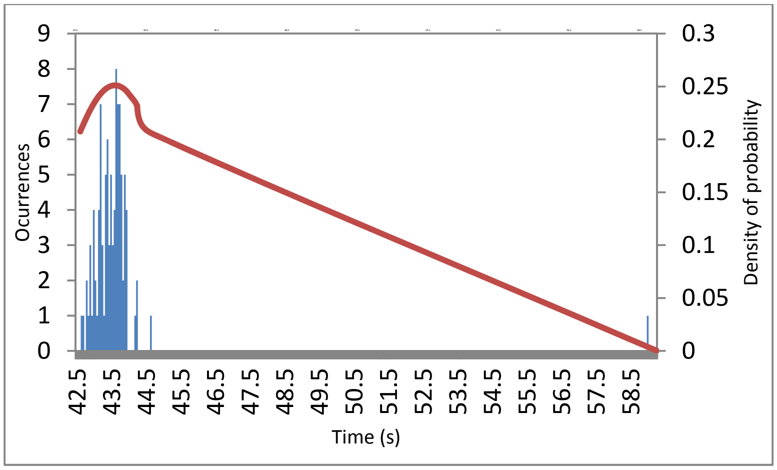


Figure 13 : *Model's run time data distribution and normal curve in 64-bit.*

The data distribution shown in Figure 13 is less obviously normal, mainly due to the higher previously noticed irregularity. However, by concentrating the analysis to the area covering the main data, a bell shape curve can be fitted. The χ^2 test result should give us a more clear indication about the normality of the set of data, as demonstrated by the following equations.

$$H_0 : O_i = 43.6169 \quad (11)$$

$$\chi^2 = 5.7866 \quad (12)$$

In fact, even with the aforementioned irregularity, the χ^2 test value is still considerably under the critical value of $\chi^2_{.050}$. This can be explained by the low occurrence of irregularities and it confirms that this set of data may be considered as a normal distribution as well. Figure 14 illustrates the quartile distribution on the *AMdll.mdl* run time.

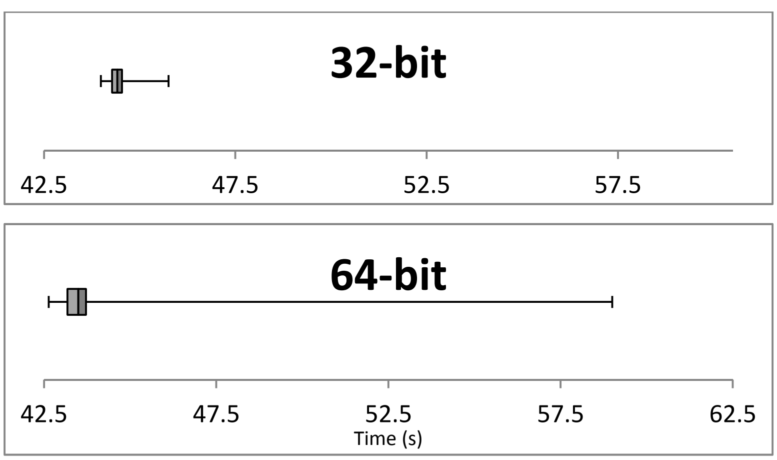


Figure 14 : *Quartile distribution on AMdll.mdl run time.*

The quartile graph of the two functions of Figure 14 shows a distribution of data that varies but that is still generally similar. The maxima are again very different but, as seen previously, mainly due to irregularities of low occurrence, where they should not be considered as a key factor. The χ^2 test will be run on the data with two different null hypotheses:

$$H_{01}: O_{i64} = O_{i32} \text{ and } H_{02}: O_{i64} = Mean_{32} \quad (7)$$

$$\chi^2_{01} = 7.4673 \text{ and } \chi^2_{02} = 7.2035 \quad (8)$$

Both χ^2 test values are smaller than the critical value of 124.342. The two null hypotheses are confirmed, and one can assume that the two set of data can be considered as equivalent, showing no significant performance variation in MATLAB for the model run time.

3.2.2.3 Interception scenario run time *(ActiveMissile.mdl)*

After the confirmation of general parameter initialisation, and the model running time, a test was also required to compute the model usage in a MATLAB scenario such as the scenario provided in the *ActiveMissile.mdl*. In that vein, Figure 15 illustrates the set of data compiled for the scenario run time in 32-bit.

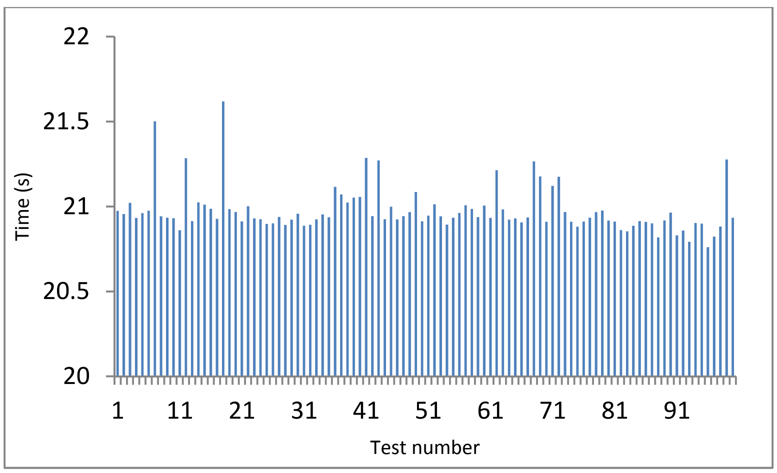


Figure 15 : *Set of data compiled for the interception scenario run time in 32-bit.*

As shown in Figure 15, the data seems to show a few more irregularities, but in general the behavior overall is stable.

Table 8 lists the main statistical values for the 100 runs on the 32-bit *ActiveMissile.mdl* run time.

Table 8 : *Main statistical values for the 100 runs on the 32-bit ActiveMissile.mdl run time.*

ActiveMissile.mdl run time 32-bit	
min	20.7602
q1	20.9110
median	20.9365
q3	20.9899
max	21.6186
mean	20.9762
sd	0.1324

From Table 8, the statistical values seem to indicate a symmetrical distribution with a fairly concentred set of the data as well as a slightly positive skewness. The overall data sample represents more or less a Gaussian distribution, as illustrated in Figure 16.

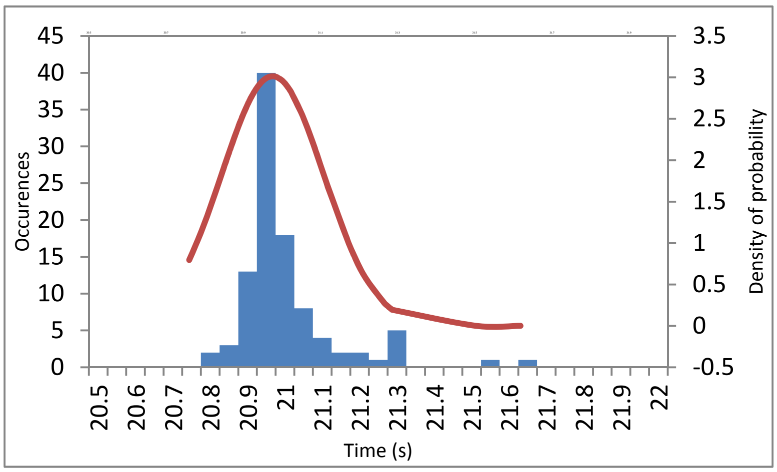


Figure 16 : *Interception scenario run time data distribution and normal curve 32-bit.*

As shown in Figure 16, the assumption of Gaussian distribution with statistical data is confirmed by the graph and its corresponding bell shaped curve. Accordingly, the χ^2 test confirms the same result.

$$H_0: O_i = 20.9762 \tag{09}$$

$$\chi^2 = 0.0835 \tag{10}$$

The χ^2 test value is significantly smaller than the critical value of $\chi^2_{.050} = 124.342$, hence confirming the normality of this sample of data. Figure 17 illustrated the set of data compiled for the interception scenario run time in 64-bit.

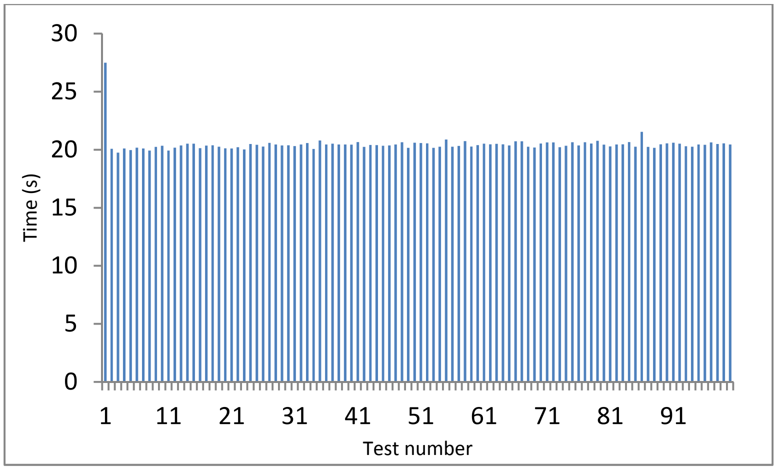


Figure 17 : *Set of data compiled for the interception scenario run time in 64-bit.*

Figure 17 shows a very constant value with very little variation for almost each and every run for the 64-bit test. Again, an irregularity appears on the first run. Table 9 lists the main statistical values for the 100 runs on the 64-bit *ActiveMissile.mdl* run time.

Table 9 : *Main statistical values for the 100 runs on the 64-bit ActiveMissile.mdl run time.*

ActiveMissile.mdl run time 64-bit	
min	19.7474
q1	20.2576
median	20.4280
q3	20.5312
max	27.5022
mean	20.4738
sd	0.7446

Statistical data listed in Table 9 shows a slightly unevenly distributed data around the median. However, the distribution seems to be representative of a normal distribution curve. Figure 18 presents the interception scenario run time data distribution and normal curve in 64-bit.

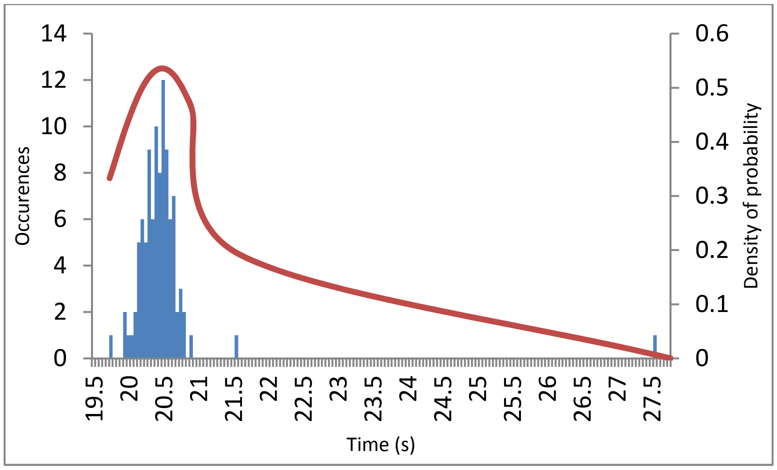


Figure 18 : *Interception scenario run time data distribution and normal curve in 64-bit.*

The graphical data presented in Figure 18 provides additional information showing a more or less Gaussian distribution with some deformation due to some irregularity within the data. Except for the irregularity, the distribution seems to be normally distributed, which is easily confirmed by the χ^2 test:

$$H_0: O_i = 20.4738 \tag{11}$$

$$\chi^2 = 2.7082 \tag{12}$$

Although the χ^2 test value is somewhat higher than the in the 32-bit test, it is still significantly smaller than the critical value of $\chi^2_{.050} = 124.342$, hence the normality of this sample can also be assumed and samples can be compared to each other, as illustrated in Figure 19.

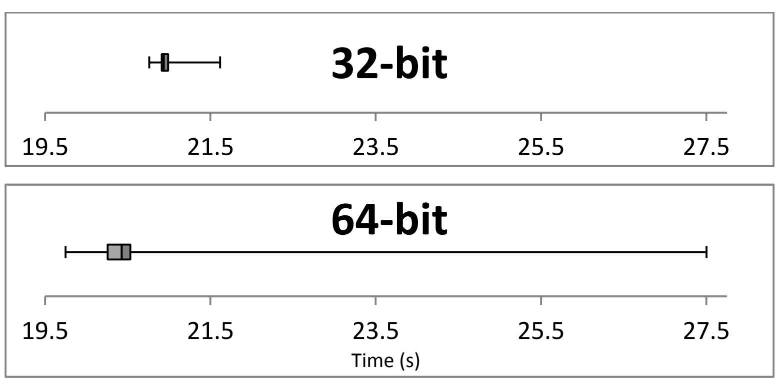


Figure 19 : *Quartile distribution of interception scenario run time.*

The quartile graphs of Figure 19 show that the 64-bit data distribution is somewhat more spread-out than the 32-bit, hence, for the primary data (between Q1 and Q3), validation is required to confirm if the data sets can be treated as equivalent. The sets of data are tested thru the double null hypotheses, as follows:

$$H_{01}: O_{i64} = O_{i32} \text{ and } H_{02}: O_{i64} = Mean_{32} \tag{13}$$

$$\chi^2_{01} = 3.9440 \text{ and } \chi^2_{02} = 3.8267 \tag{14}$$

Despite the initial assumption based on the graphical analysis, both χ^2 test values are smaller than the critical value and confirm that there is no statistically significant difference between the two samples of data. This implies that the system migration to 64-bit in Matlab is not showing performance variation for the given scenario run time.

3.2.2.4 Model compilation time *(AMdll.mdl generating the .dll file)*

The final step before a given DRDC Simulink model can be used in the SADM environment as an external component is to test the model compilation using the Simulink coder compiler (generation of a corresponding dynamic link library file (.dll)[2].

It should be noted that this portion of our tests were done on a different platform set-up (virtual machine) due to licence management limitations. Figure 20 shows the set of data compiled for the model compilation time in 32-bit.

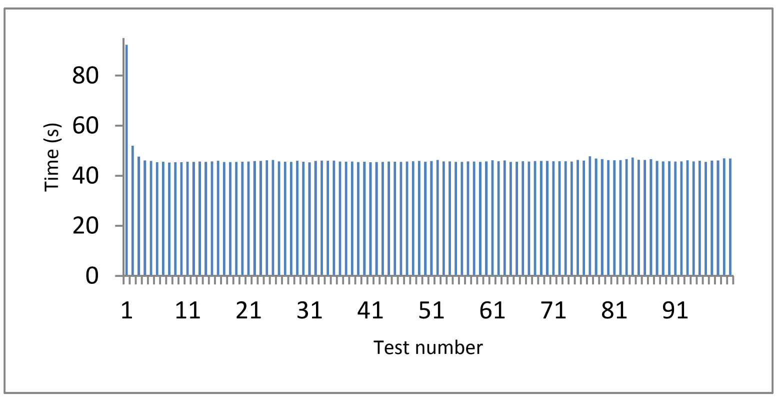


Figure 20 : *Set of data compiled for the model compile time in 32-bit.*

Compiling the model with the 32-bit set-up resulted in a very consistent set of data with a run step outside of the average range of the sample. However, the overall set of data seems constant enough to expect a normally distributed set of data.

Table 10 lists the main statistical values for the 100 runs on the 32-bit *Amdll.mdl* compilation time.

[2] Annexe A gives a complete procedure to generate the DLL file.

Table 10 : *Main statistical values for the 100 runs on the 32-bit Amdll.mdl compile time.*

AMdll.mdl compile time 32-bit	
min	45.3003
q1	45.6225
median	45.7632
q3	46.0824
max	92.3079
mean	46.4201
sd	4.6734

As shown in Table 10, the statistical values of our sample indicate a Gaussian distribution with the exception of the maximum. Indeed, the maximum of the sample, resulting in a large data range, may compromise the normality of the sample. An additional test should help to correctly define the nature of the distribution. Figure 21 presents the model compilation time data distribution and normal curve in 32-bit.

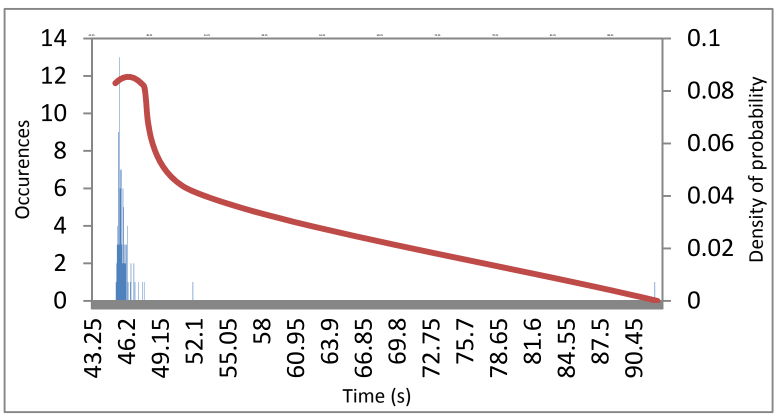


Figure 21 : *Model compile time data distribution and normal curve in 32-bit.*

The given data distribution and the normality curve of this sample (shown in Figure 21) raises questions on the normality of the curve. However, the odd shape of the normality curve seem to be mainly due to the singular irregularity. A zoom of the main area of the distribution should give a better idea of the real distribution curve shape, as shown in Figure 22.

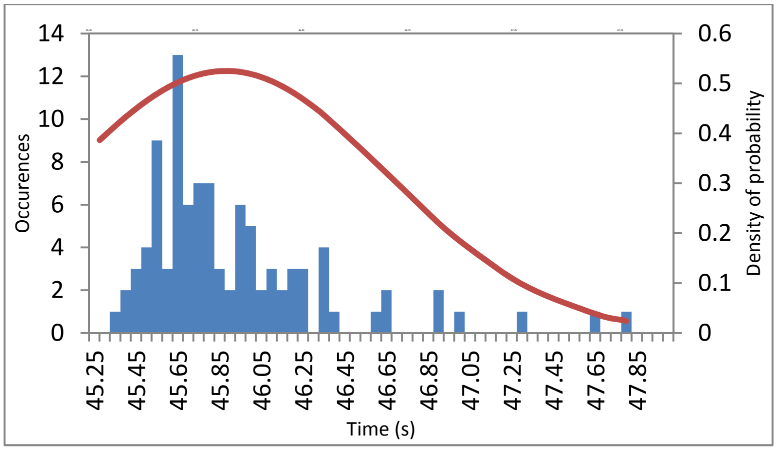


Figure 22 : *Model compile time, data distribution and normal curve 32-bit (zoomed).*

Indeed, Figure 22 shows that a close-up on the main range of data now clearly demonstrates the normality of the distribution. The χ^2 test is performed to confirm this assumption:

$$H_0: O_i = 46.4200 \tag{15}$$

$$\chi^2 = 47.0511 \tag{16}$$

The χ^2 test value is still under the critical value of $\chi^2_{.050} = 124.342$, however it is substantially higher than the previous observed values, showing that the normality of the distribution is less obvious. An easy way to evaluate that the cause of this abnormally and the resulting high χ^2 test value, is simply to remove the data suspected of creating the problem (the maxima). With this high value removed, the χ^2 test result produces a value of 2.4360, which is in a range similar to the previous samples. It confirms that the odd distribution of the sample was mainly due to the single irregularity and that the sample can be considered as normally distributed.

Figure 23 shows the set of data compiled for the model compilation time in 64-bit.

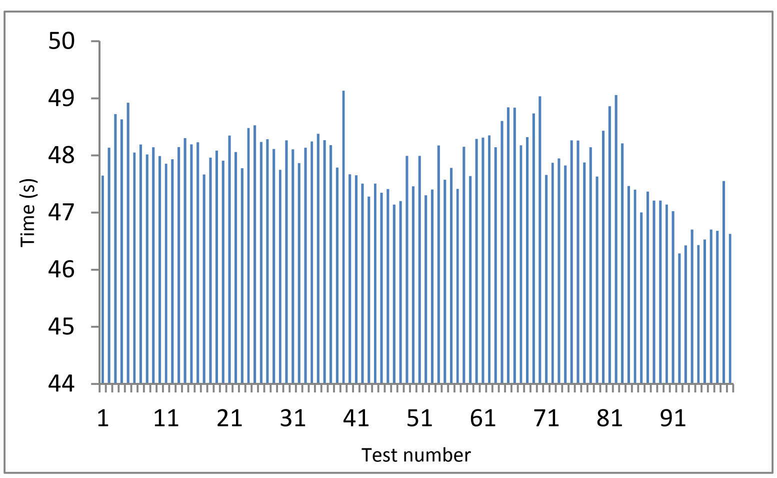


Figure 23 : *Set of data compiled for the model compile time in 64-bit.*

As shown in Figure 23, the 64-bit compiled model does not present a set of data with disproportionate values; however, the overall variation in the timing appears to be a little peculiar and cannot be automatically associated to a normal distribution. Table 11 lists the main statistical values for the 100 runs on the 64-bit *Amdll.mdl* compilation time.

Table 11 : *Main statistical values for the 100 runs on the 64-bit Amdll.mdl compile time.*

AMdll.mdl compile compilation time 64-bit	
min	46.2846
q1	47.4974
median	47.9917
q3	48.2622
max	49.1350
mean	47.8778
sd	0.6117

As shown in Table 11, a quick analysis of the main statistical values shows a globally symmetrical distribution of data, which appears to be consistent with a Gaussian distribution. The analysis of the detailed distribution as well as the χ^2 test should confirm the assumption of a Gaussian distribution. Figure 24 illustrates the model compilation time data distribution and normal curve 64-bit.

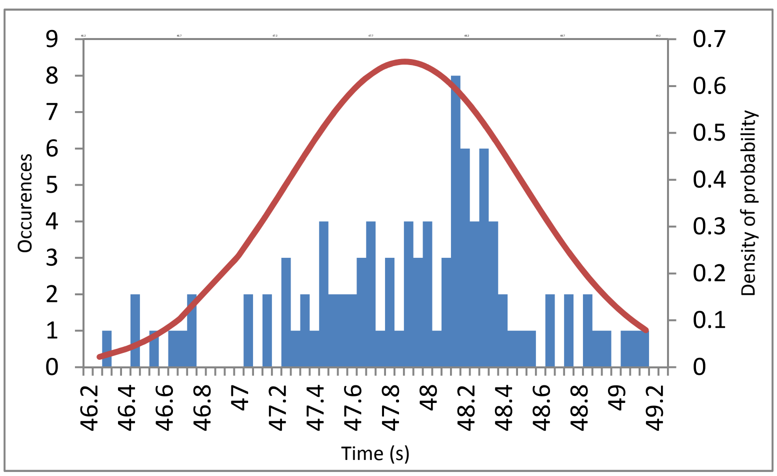


Figure 24 : *Model compile time data distribution and normal curve in 64-bit.*

The normality of the distribution is presented in Figure 24, where the bell shape of the graph shows a typical normal distribution shape. The χ^2 test result should confirm this:

$$H_0: O_i = 47.8778 \quad (17)$$

$$\chi^2 = 0.7817 \quad (18)$$

The χ^2 test value is significantly under the critical value of $\chi^2_{.050} = 124.342$, confirming the normal distribution assumption and allowing the comparison of both samples, as illustrated in Figure 25.

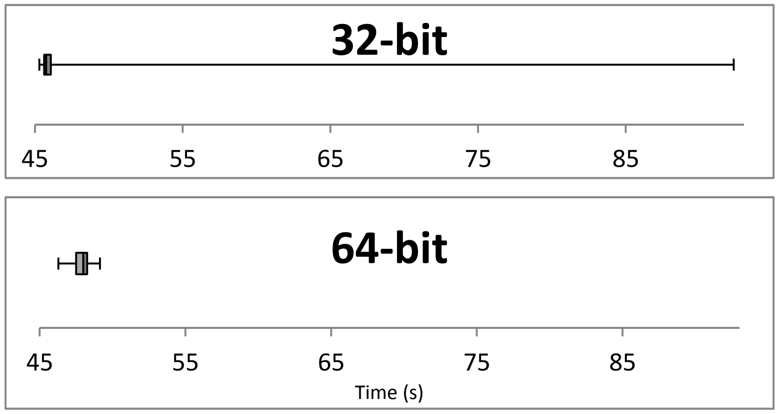


Figure 25 : *Quartile distribution of model compilation time.*

The quartile graphs of Figure 25 show that the data distribution isn't that similar at first glance. Indeed, this time the 32-bit data has a larger spread. Again, it is mainly due to a low occurrence irregularity. For the main data distribution, the two tests show a slight difference in the distribution that requires a confirmation of equivalence using the double null hypotheses:

$$H_{01}: O_{i64} = O_{i32} \text{ and } H_{02}: O_{i64} = Mean_{32} \quad (19)$$

$$\chi^2_{01} = 31.5839 \text{ and } \chi^2_{02} = 5.3840 \quad (20)$$

Both χ^2 test values are smaller than the critical value and confirm that there is no statistically significant difference between the two samples of data. This means the system migration to 64-bit is not showing a statically significant performance variation in MATLAB for the model compile time.

3.2.3 Conclusion

Each and every test realized on the MATLAB software shows no statically significant improvement or decrease of passing to a 64-bit operating system. The overall performance results show similar characteristics and after reviewing each test result one can conclude that the 64-bit software is performing in a similar way to the 32-bit version.

The fact that the 64-bit version does not result in a performance gain is not surprising. In fact, current software coding is still based on a 32-bit architecture that does not totally make an optimal use of the capabilities of a 64-bit system. The conservation of performances was also announced by MathWorks [4]. Finally, it is very important to mention that: **MathWorks is also issuing a stability warning concerning the usage of MATLAB for computing on sensitive applications**[3]. Indeed it is recommended that high sensitivity calculations should be compared on both operating systems because the 64-bit set-up may use CPU registers differently with data of

[3] http://www.mathworks.com/products/matlab/preparing-for-64-bit-windows.html

type single which may lead to minor data variations. It is worth noting that this fact might explain the difference in Ps previously observed on the SADM scenario results.

3.3 SADM

The main MATLAB and Simulink components of a DRDC library model have been passed through the performance testing to validate their performance on a 64-bit set-up. It showed no gain or decrease in performance or instability with this set-up. However, the components of the DRDC library are often used in external software such as SADM. In the particular case of SADM, this software works in collaboration with MATLAB and as explained before, the 64-bit migration of MATLAB forces a migration to a 64-bit version of SADM. The stability of the simulation has already been demonstrated previously, but since SADM simulations are often run in large and time consuming batches, a closer look will be taken to investigate the performance of the 64-bit version of that software.

3.3.1 Version and methodology

The test will be performed with the same versions of the software previously used, namely:

- SADM version 5.3.0.1 (32 and 64 bits)
- MATLAB R2013b (8.2.0.701) (32 and 64 bits)

For the simulations, the *ActiveMissile.scn* will be the test scenario and it will be executing the external user missile *ActiveMissile.wpn,* which will call external user parameters by a call of the previously generated DLL[4] file.

The methodology of validation will be quite similar to the one previously used (see Section 3.2.1). A simple MATLAB program will be created to initiate batches of 100 simulations with the specified components and timing which will be stored for each simulation. In addition to the general scenario performance, an additional check will be performed on the effects of performance caused by changing the number of MC trials for a given scenario.

3.3.2 General scenario performance, 32 vs 64 bits

The first set of simulations was composed of a batch of 100 subsequent simulations with only 1 MC trial and will serve as the baseline performance comparison. Figure 26 illustrates the set of data compiled for a SADM scenario execution time in 32-bit.

[4] Annexe A gives a complete procedure to generate the DLL file.

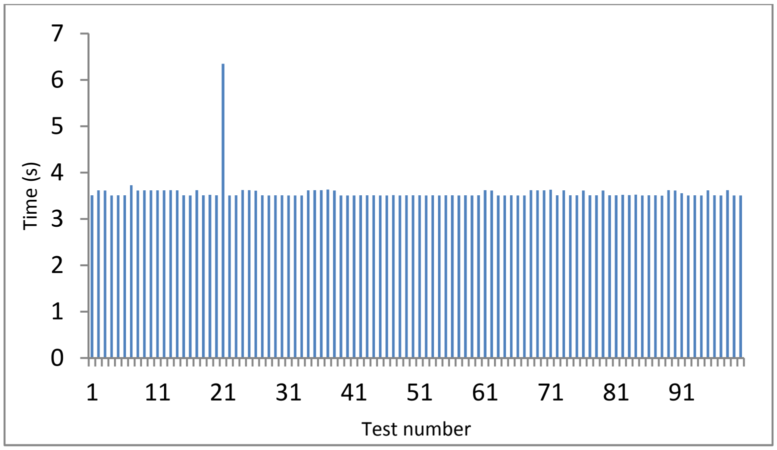


Figure 26 : *Set of data compiled for SADM scenario execution time in 32-bit (1MC).*

As shown in Figure 26, the collected data appears to be relatively constant, except with one observed irregularity. The overall behavior of the computing performances seems to comply in regularity with the previously observed phenomena. Once again, it is likely to observe odd normal distribution curves due to irregularities, but the regularity of the rest of the data is a good indication of a Gaussian distribution. Table 12 lists the main statistical values for the 100 runs on SADM 32-bit scenario run.

Table 12 : *Main statistical values for the 100 runs on SADM 32-bit scenario run.*

SADM Scenario 1 MC 32-bit	
min	3.5042
q1	3.5086
median	3.5106
q3	3.6159
max	6.3492
mean	3.5747
sd	0.2839

As shown in Table 12, the scenario run time statistical values show a very high concentration of data to the exception of the maxima observed. Figure 27 presents the SADM scenario execution time data distribution/ normal curve in 32-bit.

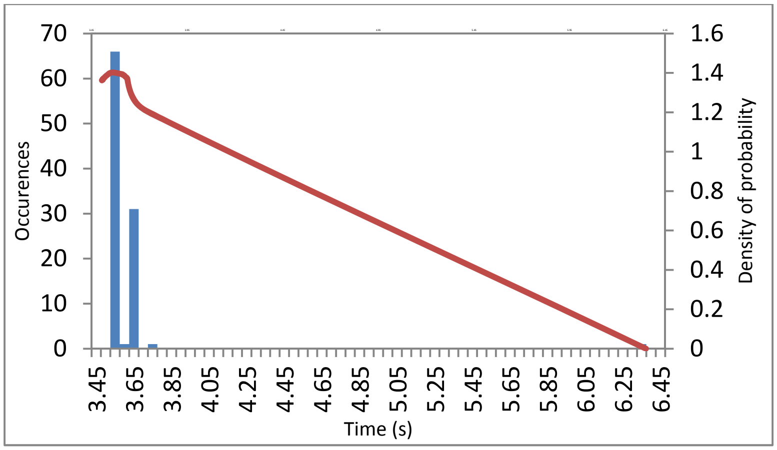


Figure 27 : *SADM scenario execution time data distribution/ normal curve in 32-bit (1MC).*

As expected, it is difficult to determine if Figure 27 presents a normal distribution due to the deformation introduced by the irregularity. Indeed, the section in the area of the main data seems to be bell shaped, but the χ^2 test should easily confirm the normality of the distribution, as follows:

$$H_0 : O_i = 3.5747 \quad (21)$$

$$\chi^2 = 2.2554 \quad (22)$$

The χ^2 test value is significantly less than the critical value confirming the normality of the data distribution. Figure 28 illustrates the set of data compiled for SADM scenario execution time in 64-bit.

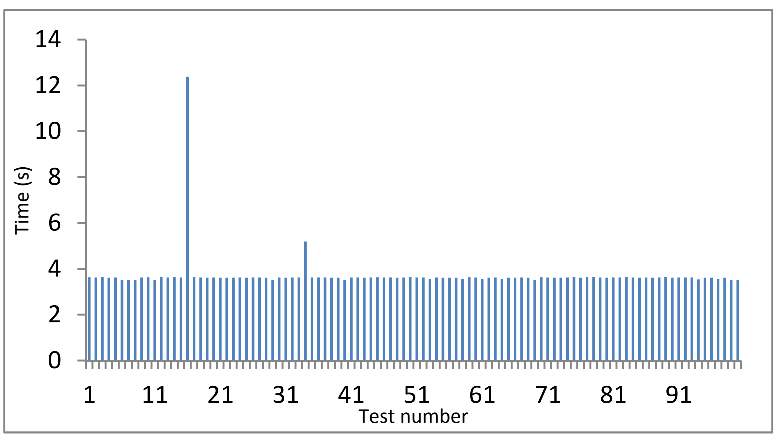


Figure 28 : *Set of data compiled for SADM scenario execution time in 64-bit (1MC).*

As shown in Figure 28, the 64-bit results have a very similar distribution pattern: high data regularity and low occurrence irregular data. This first glimpse of the distribution is obviously promising for a Gaussian distribution. Table 13 lists the main statistical values for the 100 runs on SADM 32-bit scenario run.

Table 13 : *Main statistical values for the 100 runs on SADM 64-bit scenario run.*

SADM Scenario 1 MC 64-bit	
min	3.5068
q1	3.6161
median	3.6183
q3	3.6222
max	12.3817
mean	3.71028
sd	0.8864

Here again, the statistical data of Table 13 show a very high concentration of data with a very distant maxima data point. Figure 29 presents the SADM scenario execution time data distribution/normal curve in 64-bit.

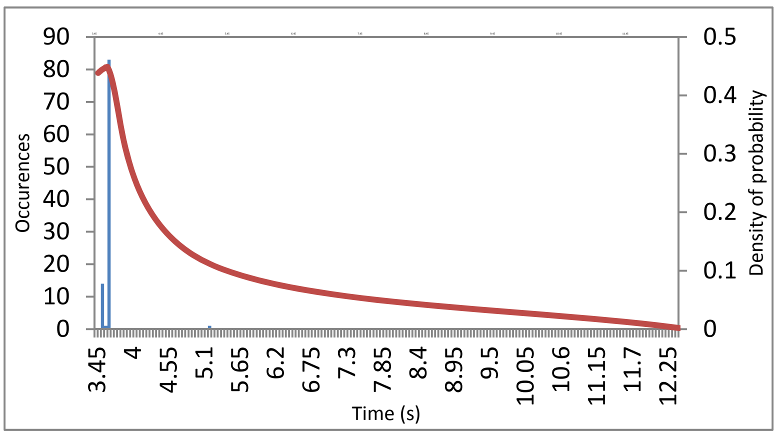


Figure 29 : *SADM scenario execution time data distribution/normal curve in 64-bit (1MC).*

Again, Figure 29 shows that the normal curve is offset by the low occurrences at the end which makes the curve look more like a logarithmic distribution. The χ^2 test should give a better indication about the normality of the distribution:

$$H_0: O_i = 3.7103 \quad (23)$$

$$\chi^2 = 21.1745 \quad (24)$$

The χ^2 test value is under the critical value, a bit higher than usual, but the aforementioned observed maximum is mainly responsible for the offset, as it has been demonstrated. Nevertheless, the test result confirms the normality of the data distribution, allowing a comparison between samples, as illustrated in Figure 30.

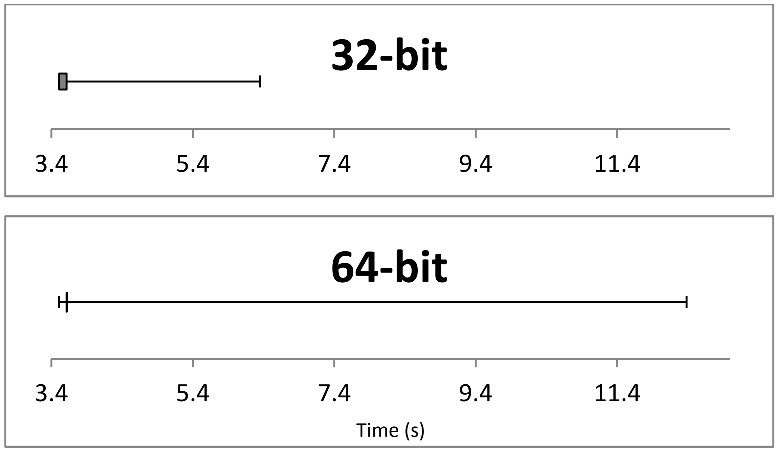


Figure 30 : Quartile distribution for *SADM scenario execution time (1 MC).*

As shown in Figure 30, except for the maxima, the data distribution seems to be very similar between the 32 and 64 bits trials. Again the level of comparison will be judged by the chi-square test on the same two null hypotheses, as follows:

$$H_{01}: O_{i64} = O_{i32} \text{ and } H_{02}: O_{i64} = Mean_{32} \quad (25)$$

$$\chi^2_{H01} = 24.5326 \text{ and } \chi^2_{H02} = 22.4918 \quad (26)$$

Both χ^2 test values appear to be higher than the usual previously observed test results. The higher value is mainly, again, due to the maxima. If the maxima are suppressed, the value is less than 1. Despite this observation, the test values are smaller than the critical value and confirm that there is no statistically significant difference between the two samples of data. This means the system migration to 64-bit is not showing statically significant performance variation for the SADM software as well.

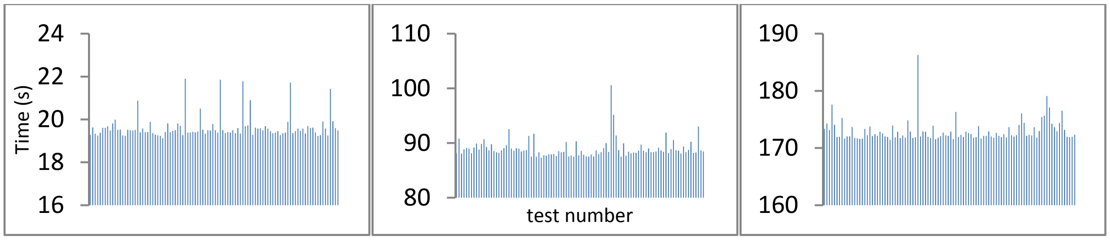
3.3.3 Impact of the number of Monte Carlo trials used

Given that the general simulation has proved to be as stable and performant as the 32-bit system, it was decided to increase the number of MC trials within a given scenario. This approach was studied since the 64-bit system has been proven to be more efficient especially when computing with a large amount of data. Since SADM simulations often require multiple batch runs of a single scenario, knowing how the number of MC trial is influencing the performance of our system was obviously relevant.

Now that we already know that the general computing time of both the 32 and 64 bits set-up can be considered as normally distributed, the data for the MC trials test will be summarised.

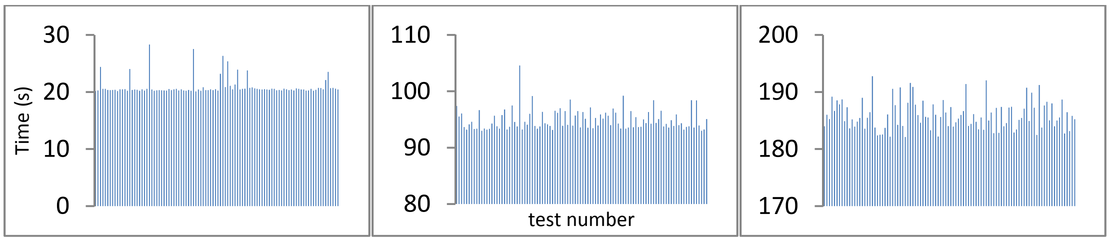
3.3.3.1 Data gathered

Figure 31 and Figure 32, respectively, illustrate the data compiled for SADM scenario execution times of 32 and 64 bits with MC variations.



a) 10 MC *b) 50 MC* *c) 100 MC*

Figure 31 : *Data compiled for SADM scenario execution time in 32-bit with MC variations.*



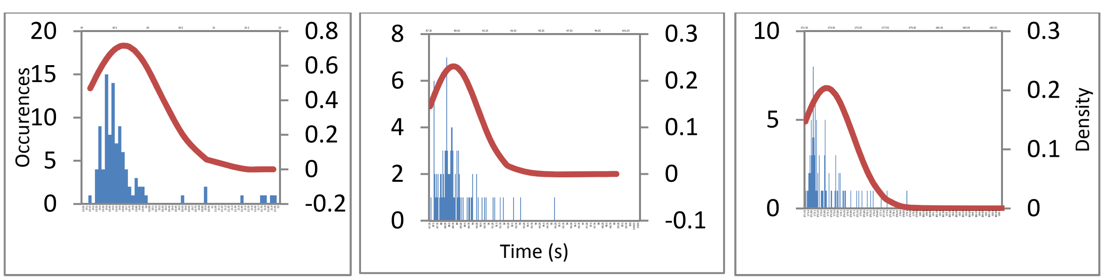
a) 10 MC *b) 50 MC* *c) 100 MC*

Figure 32 : *Data compiled for SADM scenario execution time in 64-bit with MC variations.*

Through inspection of the raw data illustrated in Figure 31 and Figure 32, specific information that can be assembled. First, all sets of data, even with some irregularities, can probably be considered as normally distributed. Another observation is that as the number of MC trials increase, the instability (noise) it generates also increases. As it was previously stated, this is logical because each trial has its level of randomness and the more that is added, the more variation is likely to be observed. Finally, the last information from the graphs pertains to the non-direct linearity of the computing time. Indeed, for the 32 and 64 bits results, the time to execute 100 MC trials is not a factor of 10 more than that observed for 10 MC runs.

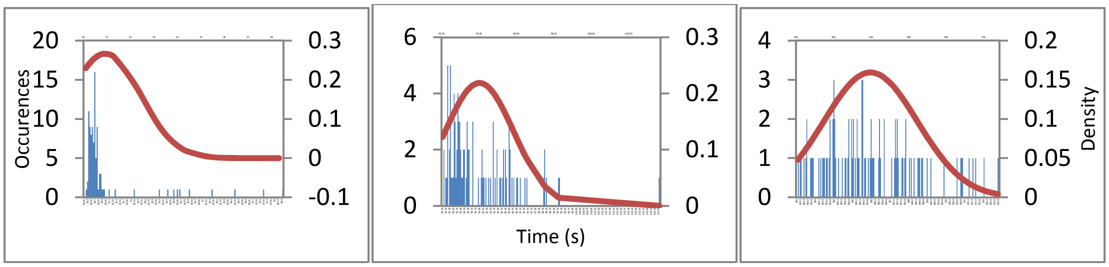
3.3.3.2 Data distribution and normality

Figure 33 and Figure 34 respectively present the SADM scenario execution time data distribution and normal curve in 32 and 64 bits.



a) 10 MC *b) 50 MC* *c) 100 MC*

Figure 33 : *SADM scenario execution time data distribution and normal curve 32-bit.*



a) 10 MC *b) 50 MC* *c) 100 MC*

Figure 34 : *SADM scenario execution time data distribution and normal curve 64-bit.*

The data distribution and normal curve graphs show that each sample of data has a characteristic bell shaped curve. It also shows that the more MC trials that are executed, the less skewness is visible. For a Gaussian distribution, this is totally intuitive because an increase of the MC trials could be viewed as an augmentation of the sample size, which would result in a normal curve being more representative of a perfect normal distribution curve.

3.3.3.3 10 MC sample comparison

To simplify the comparison, only the χ^2 test value will be considered, remembering that the critical value of $\chi^2_{.050} = 124.342$:

- **Sample normality**

$$H_{01}: O_{i32} = Mean_{32}, \chi^2_{H01} = 1.5769, \text{Normality = ok} \quad (27)$$

$$H_{02}: O_{i64} = Mean_{64}, \chi^2_{H02} = 10.6801, \text{Normality = ok} \quad (28)$$

Both samples are confirmed to be normally distributed.

- **32 vs 64 bits**

$$H_{01}: O_{i64} = O_{i32}, \chi^2_{H01} = 20.9853, \text{Normality = ok} \quad (29)$$

$$H_{02}: O_{i64} = Mean_{32}, \chi^2_{H02} = 19.7185, \text{Normality = ok} \quad (30)$$

Both samples are not showing enough statistically significant differences to be considered different. Therefore the 64-bit set-up does not show any significant improvement in performance when ten times the amount of initial MC trials is considered.

- **Effects of the Monte Carlo trials alone**

This assessment is performed in order to verify if the execution time is directly proportional to the number of MC trials:

$$H_{01}: O_{i32(10)} = 10 \times mean_{32(1)}, \chi^2_{H01} = 726.4749, \text{Normality = not equivalent} \quad (31)$$

The time to compute of 19.6405 sec is showing a gain of 45% over the expected linear value of 35.7470 sec.

$$H_{02}: O_{i64(10)} = 10 \times mean_{64(1)}, \chi^2_{H02} = 711.8070, \text{Normality = not equivalent} \quad (32)$$

The time to compute of 20.9205 sec is showing a gain of 44% over the expected linear value of 37.1028 sec.

The χ^2 test values obtained show that our assumption about the none-direct linearity of the computing time as a function of the MC trials amount was founded. Indeed, multiplying the MC trials amount by 10 shows an increase in performance of approximately 44%.

3.3.3.4 50 MC sample comparison

- **Sample normality**

$$H_{01}: O_{i32} = Mean_{32}, \chi^2_{H01} = 3.3588, \text{Normality = ok} \quad (33)$$

$$H_{02}: O_{i64} = Mean_{64}, \chi^2_{H02} = 3.5064, \text{Normality = ok} \quad (34)$$

Both samples are confirmed to be normally distributed.

- **32 vs 64 bits**

$$H_{01}: O_{i64} = O_{i32}, \chi^2_{H01} = 47.6982, , \text{Normality = ok} \quad (35)$$

$$H_{02}: O_{i64} = Mean_{32}, \chi^2_{H02} = 43.8720, \text{Normality = ok} \quad (36)$$

Both samples are not showing enough statistically significant difference to be considered different. Therefore the 64-bit set-up is not showing any performance improvement with fifty times the amount of MC trials.

- **Effects of the Monte Carlo trials alone**

$$H_{01}: O_{i32(50)} = 50 \times mean_{32(1)}, \chi^2_{H01} = 4507.578, \text{Normality = not equivalent} \quad (37)$$

The time to compute is showing a gain of 50% over the linear value.

$$H_{02}: O_{i64(50)} = 50 \times mean_{64(1)}, \chi^2_{H02} = 4421.085, \text{Normality = not equivalent} \quad (38)$$

The increase of performance is approximatively 49% over the linear value.

The χ^2 test values obtained show that our assumption about the none-direct linearity of the computing time function of the MC trials amount was founded. Indeed, multiplying the MC trials amount by 50 shows an increase in performance on the linear value of approximately 49%.

3.3.3.5 100 MC sample comparison

- **Sample normality**

$$H_{01}: O_{i32} = Mean_{32}, \chi^2_{H01} = 2.2149, \text{Normality = ok} \quad (39)$$

$$H_{02}: O_{i64} = Mean_{64}, \chi^2_{H02} = 3.3655, \text{Normality = ok} \quad (40)$$

Both samples are confirmed to be normally distributed.

- **32 vs 64 bits**

$$H_{01}: O_{i64} = O_{i32}, \chi^2_{H01} = 102.5157, \text{Normality = ok} \quad (41)$$

$$H_{02}: O_{i64} = Mean_{32}, \chi^2_{H02} = 100.8298, \text{Normality = ok} \quad (42)$$

Both samples are not showing enough statistically significant difference to be considered different. Therefore the 64-bit set-up is not showing any performance improvement with one hundred times the amount of MC trials. However, is important to note that the chi-square value has increased noticeably with the number of MC trials.

- **Effects of the Monte Carlo trials alone**

$$H_{01}: O_{i32(100)} = 100 \times mean_{32(1)}, \chi^2_{01} = 9231.311 \text{ Normality = not equivalent} \quad (43)$$

The time to compute is showing a gain of 50% over the linear value.

$$H_{01}: O_{i64(100)} = 100 \times mean_{64(1)}, \chi^2_{01} = 9519.814, \text{Normality = not equivalent} \quad (44)$$

The increase of performance is approximatively 52% over the linear value.

The χ^2 test values obtained show that our assumption about the none-direct linearity of the computing time as a function of the MC trial amount was founded. Indeed, multiplying the MC trials amount by 100 shows an increase in performance on the linear value of approximately 50%.

3.3.3.6 Conclusion

The tests performed with a variation of the number of MC trial provided interesting information. First, it draws important conclusions regarding the computation time as a function of the number of MC trials. The first assumption was to expect a direct linear proportionality, meaning that the computing time would directly relate to the number of MC trials (e.g. 10 times more trials meaning 10 times more computing time). However, the test results showed a different phenomenon, as illustrated in Figure 35.

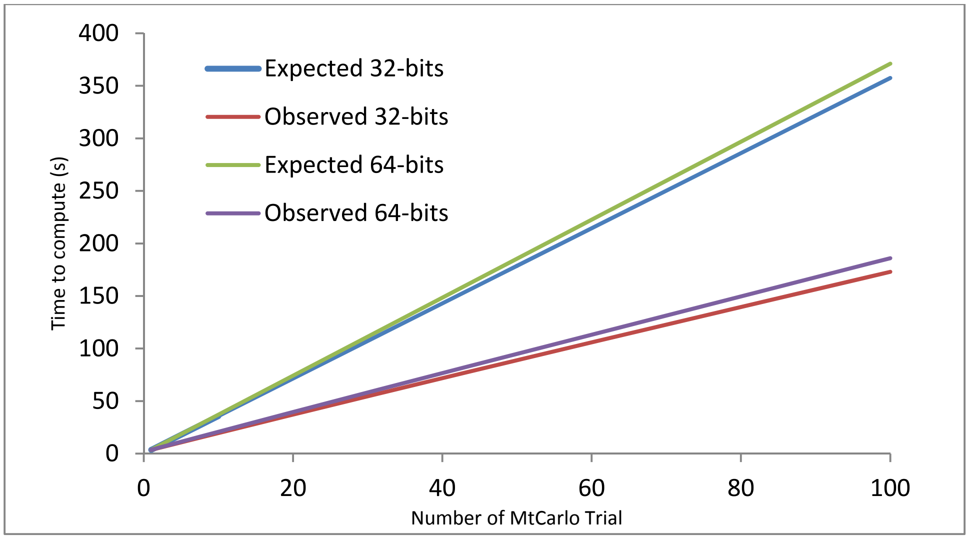


Figure 35 : *Variation on SADM computing time function of the number of MC trials.*

As shown in Figure 35, the statistical study of the data shows a quasi-linearity of the computing time function of the number of MC. Indeed, exceeding 10 MC trials, the computing time is approximately 50% of the direct linear expected value and this, for both the 32 and 64 bits set-up:

$$Ctime_{x>10MtC} = Ctime_{1MtC} \times 0.5 \tag{45}$$

If the number of MC trial is between 1 and 10, the estimation of the computing time should be considered as directly proportional since not enough data has been gathered in those experiments to define a more accurate approximation:

$$Ctime_{x<10MtC} = Ctime_{1MtC} \times Nb\_MtCarlo \tag{46}$$

As a warning, these assumptions are considered valid for tests realised in the vicinity of the MC trial number tested. The authors are not sure of the validity of such an assumption for a larger number of MC trials (>500).

The other interesting point is how the χ^2 test value is constantly growing as a function of the MC trial when comparing the 32 and 64 bits results, as illustrated in Figure 36.

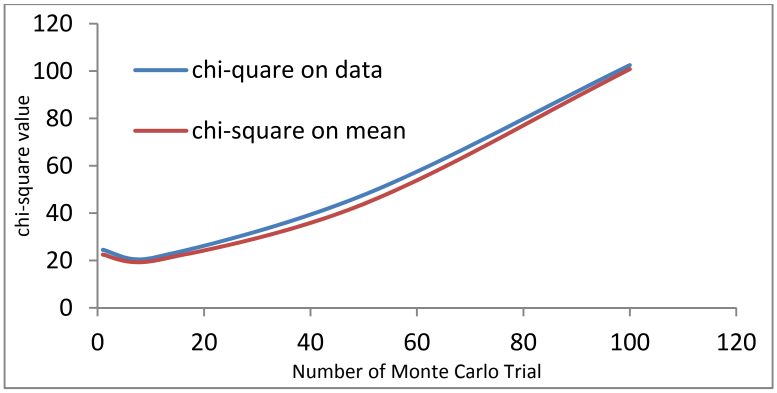


Figure 36 : χ^2 test value *on SADM computing time comparison between 32 and 64 bits.*

As shown in Figure 36, even if, on all the samples tested, the similarity of the data between the 32 and 64 bits was statically close enough to suppose an equivalence, the tendency observed on the chi-square test value graph clearly shows that for tests realised over 100 MC trials, there is a good chance that the χ^2 test value will be above the $\chi^2_{.050} = 124.342$, i.e., the critical value. This seem consistent enough with the MathWorks explanation, where the 64-bit system is suspected to shows performance gain, but mainly for the computation of larger sets of data.

However, the expected "gain" as an interpolation of the current data of the 100 MC trials shows a diminution of performance of around 7% while computing on a 64-bit system, as shown in Figure 37.

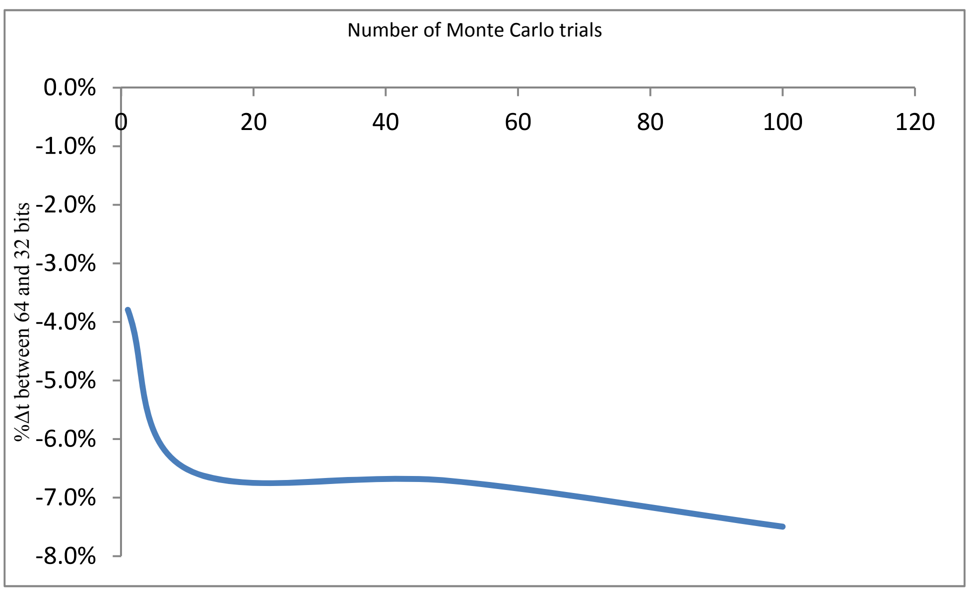


Figure 37 : *Computing time variation on SADM scenario for a 64-bit set-up compared to a 32-bit set-up.*

The variation in SADM scenario computing time with the current test results show a certain level of linearity that might indicate a tendency for the computing time on a 64-bit to increase with the number of MC trials.

However, that extrapolation should not be taken with caution, since the current analysis results are not conclusive enough to indicate a real performance loss/gain as a function of the parameters tested.

4 Conclusion

In summary, comparisons were realized to investigate the effect of 32 and 64 bit execution on SADM scenario simulation results, DRDC library model performance, parameter initialisation, on run time and on compilation time. Respective conclusions and recommendations are given below.

4.1 Stability of the software after the 64-bit migration

Overall, each and every test executed showed that the migration to a 64-bit set-up does not generate any instabilities in the software execution or computing processes. Despite the observed overall stability, it is important to mention that a slight but constant irregularity was observed on the Pk/Ps within the SADM simulations. With the current limited level of access to the internal structure of the SADM software, the explanation of the variation of those parameters is hard to investigate and thus our assumptions are more of an educated guess than based on scientific evidence. A more extensive understanding of those variations would require a more precise study using the SADM source code that would investigate the specific computed data and the various steps of the scenario simulation. For now, the nature of the differences observed is not judged significant enough to be considered as a major problem. It is however definitely a point that should be further investigated.

4.2 Performance variation of the software after the 64-bit migration

The MATLAB and Simulink test results show no performance gain or degradation related to the 64-bit set-up. All test results from the 64-bit software are statically coherent with the data obtained from the 32-bit software. The fact that the 64-bit software does not perform better than the 32-bit can probably be explained by the fact that most 64-bit software are still working with an important core that was initially designed for a 32-bit system. Performance improvement within the next ten years may be observed when the original core structure of the software will be updated to maximize the 64-bit full potential especially under a Windows exploitation system. DRDC should be aware that the current tests, even if they show no statically significant performance variation, are giving information indicating a decrease in performance for a 64-bit set-up for a scenario with more than 100 MC trials. This does not mean that tests performed with large batches of MC trials should be discarded. It just shows that those 64-bit tests might take a little longer to compute. The reason for this loss of performance is counterintuitive with the literature on 64-bit systems. Again, the exact cause of that performance loss is hard to explain but a more precise study using the SADM source code could be done to determine if the problem comes from SADM.

4.3 Others significant information

During the tests that focused on the number of MC trials, there was a performance variation observed that was not relevant to the bit set-up of the software itself, but directly related to the number of MC trials. It has been demonstrated that while adding MC trials to a simulation, on a 32 or 64 bits set-up, the overall computing time for a given scenario implies a relative time/MC trials ratio that is not constant. Finally, BAE is currently implementing a new weapon interface that uses a different methodology to call an external user missile. At the moment of the realization of the current study, the interface was not compatible with the current DRDC model. When the compatibility of this interface will be achieved, a similar set of tests should be performed to ensure that the new interface is producing similar results within an acceptable performance range.

References/Bibliography

[1] Altman D.G., Bland J.M.. *Statistics notes: the normal distribution*, 1995; p.298

[2] Grinstead, C. M., Snell, J. L., *Introduction to probability*, 1998; p.325

[3] Cramer, D., *Fundamental statistic for social research*, London, 1998

[4] http://www.mathworks.com/products/matlab/preparing-for-64-bit-windows.html

This page intentionally left blank.

Annex A User guide for DLL file generation

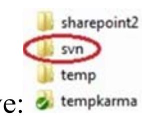
1. Requirements:

    a. MATLAB R2013b and Simulink with a Simulink Coder licence;

    b. A C++ compiler supported by Simulink Coder (e.g. Microsoft Visual Studio C++ 2010 Pro SP1 or Microsoft Windows SDK 7.1);

    c. Tortoise SVN;

2. Setup/ensure that a SVN folder is accessible by your computer

    a. The folder should be located on your “C” or “D” drive: 

;

        i. If you have a SVN folder, make sure it includes the “WeaponsSystemsLibrary” folder;

    b. If the “SVN” or “WeaponsSystemsLibrary” folders are not present:

        i. In the desired path, create a new folder named “SVN”;

        ii. In the “SVN” folder, create the WeaponsSystemsLibrary” folder;

        iii. Right-click on the created folder then select “SVN Checkout…”;

        iv. In the URL of repository field you should have:

        https://val-a-sa-svn/svn/WeaponsSystemsLibrary

        v. Click “ok” and confirm your identity with your computer credential;

        vi. Wait for the commit operation, it might take several minutes;

3. Open the MATLAB software;

4. Set the required path using the set path button;

Then, choose the option “add with subfolders”;

Choose the previously created “WeaponsSystemsLibrary” folder;

Click “Save” than “Close”;

5. In the MATLAB command screen type: “run InstallWeaponSystemsLibrary.m” then enter;

    a. Choose the “Clean Install” option (1);

    b. “Do you want to build the MEX files ?”, choose “Yes” (2);

    c. “Do you want to set up a default compiler” choose “Yes” (y);

    d. “Would you like mex to locate installed compilers ?” choose “Yes (y);

    e. “Select a compiler”, choose a C++ compiler supported by Simulink Coder (1);

        i. If this option is unavailable, check that you have a C++ compiler supported by Simulink Coder installed on your computer.

    f. To “Please verify your choices:” if you’ve made the correct choice confirm it (y);

    g. Wait for the generation of the MEX files;

    h. To “Do you want to build the LIB files (optional, requires Matlab Coder)?”, choose Yes (2);

    i. Wait for the LIB files to be created;

6. Do as suggested and type “RunAllSanityTests”;

    a. Wait for the tests to be completed;

    b. All tests should be passed successfully;

7. Open the desired weapon model (models should be located in the “WeaponSystemsLibrary” folder under various path depending of the model;

8. Type: “load ActiveMissileData.mat” and press “enter”;

    Wait for the data to be initialized;

9. Type: “run ActiveMissileGenerateParameters.m” and press “enter”;

    Wait for the parameters to be initialized;

10. Click on the “build” button ;

    a. If the build button is not there or inactive, Simulink coder might not be available on you station, contact your IT support responsible.

    b. Otherwise, allow the Simulink coder to compile the model;

11. Go to your MATLAB working directory;

12. Look for a new folder* that has been created:

***Note**: You folder name may differ from that specific name, most of the time the folder will be named “model_name”_wes_malloc_rtw.

13. The previously mentioned folder should contain your specific .dll file.

This page intentionally left blank.

List of symbols/abbreviations/acronyms/initialisms

CF	Canadian Forces
DF	Degrees of freedom
DND	Department of National Defence
DRDC	Defence Research and Development Canada
DSTKIM	Director Science and Technology Knowledge and Information Management
LSA	Librairie Systèmes d'Armes
MC	Monte Carlo
Pk	Probability of kill
Ps	Probability of survival
SADM	Ship Air Defense Model
VM	Virtual Machine
WS	Weapons Systems
WSL	Weapon System Library