



## A Directly Programmed Solution to ARGESIM Benchmark C8 'Canal-and-Lock System' using Visual Basic for Applications (VBA)

Thomas Löscher, Josef Mosser, TU Wien, Austria  
*thomas@loescher.at, e0126655@student.tuwien.ac.at*

SNE 17/3-4, December 2007

**Simulator:** Microsoft Excel is a spreadsheet application that provides simple to advanced means of creating and managing any type of list. To even make it more powerful and production oriented, MS Excel, like all members of the Microsoft Office family, ships with a computer language and a programming environment. Microsoft Visual Basic is one of the easiest programming languages to get started with for budding programmers. It has a very complete object model, easy to understand syntax (based around several flavours of traditional BASIC), and can provide Windows style interaction with the user. The Visual element in Visual Basic comes from the point and click interface - it is programmed visually, at least at the user interface level. Actions (scripts, macros) are attached to specific parts of the user interface. These are executed or evaluated at the time that the user interacts with the software. The general approach makes it very quick and easy to put together an application, and while the result might suffer from some performance issues, it is a great platform if performance is not the most important factor. Where VB comes into its own, however, is when it is integrated with other Microsoft applications. At some point, Microsoft introduced Visual Basic for Applications. This provided the power of the VB language coupled with the object model of the Office Suite. In other words, each Office component could be interacted with in a way that provided functionality beyond simple macro recording. Put another way; where users could record key presses, menu choices, and mouse movements and so on, now they could also write little programs to interact with the application in a conditional manner. In fact, VBA brings all the power of a programming language to the macro environment.

**Model:** The model represents a so called canal and lock system where barges move from east to west and vice versa. On their way through the channel the barges has to pass a lock where the water is raised and lowered respectively if it is necessary. The used channel is very narrow and therefore barges can pass the channel only one after another in each



Figure 1. Class modules

direction. The defined system configuration leads to special control rules for the use of the channel and the lock.

**A-Task:** The model is implemented in an object-oriented way. Therefore the whole power of Visual Basic is used to define class modules for the needed objects. Figure 1 shows the four used classes: Source, Queue, Server and Sink. In these classes several properties are de-fined. An own event handler is programmed to model and simulate the behaviour of the barges, the lock and the channels. Macros are implemented to control the flow of the barges and to define and to change parameters and initial conditions.

For the input of the different parameters a simple user interface is created to give the user the ability to change the different input values of the Eastbound and Westbound barges. The produced simulation results are collected on special Excel spreadsheets and after the simulation Excel and VBA built-in functions are used to derive the mean, standard deviation and confidence intervals. Figure 2 shows a simple visualisation of the simulation model. The changing

	Westbound Barges finished		"empty" Lock		Eastbound Barges finished	
	183				195	
	197	2			11	195
Eastbound Barges created	Eastbound Barges Queue	West Canal	Lock	East Canal	Westbound Barges Queue	Westbound Barges created

Figure 2. Visualisation



of the colours from green to red realise the current state of the lock and the channel. Further interesting values like simulation time, Queue contents or created barges are changing during the simulation run. For the experiments the visualisation can be disabled to increase the speed of the simulation.

**B-Task:** In this task the model is validated through five different sets of deterministic data. The changing of the data is easily possible through the small user interface. All five data sets lead to the correct results.

**C-Task:** In task c different variance reduction experiments should be done. Therefore the inter-arrival time of the barges is set to an exponential distribution with an expected value of 75 minutes. For this purpose uniform distributed values are created on a special spreadsheet before the simulation is starting. During the simulation these uniform distributed values are used one after another. The selection of the values is controlled by two global counters which are increased during the simulation. This approach is chosen because of the simple use and selection of random numbers for the experiments with antithetic variates and common random numbers. For each pair the same random numbers are used. With the VBA and Excel built-in function "Randomize" the seed of the random number generator is changed and values for a new independent run can be created. The inverse function of the exponential distribution is not implemented as built-in function in Excel and VBA. For this reason the inverse function is directly programmed and the uniform distributed values are used to calculate and create exponential distributed values.

Table 1 shows the results of the average barge transit time for three independent experiments on a pooled basis of 100 independent simulation runs. Column 3 and 4 show the length of the confidence interval and the standard deviation of each run. The parameters "Eastmax" and "Westmax" are set to 5 in both cases.

Table 2 shows the results for the same experiment with antithetic random variates. The use of this variance reduction method leads to a reduction of the

	Mean	CI	Std.	Reduction
Run 1	411,2	+/-24,0	103,0	41,2%
Run 2	415,9	+/-21,6	92,7	46,5%
Run 3	408,5	+/-18,6	80,1	60,4%

**Table 1.** Results of Antithetic Random Variates

	Mean	CI	Std.
Run 1	391,8	+/-40,8	175,4
Run 2	400,4	+/-40,3	173,3
Run 3	397,2	+/-47,0	202,2

**Table 2.** Results of 100 independent replications

	Mean	CI	Std.
Run 1	-40,4	+/-63,7	273,9
Run 2	39,3	+/-66,7	286,9
Run 3	50,0	+/-64,7	278,1

**Table 3.** Difference of 50 pairs

confidence interval length which can be seen in column 5 of Table 1.

Table 3 shows the results for the difference of the mean for "Eastmax" and "Westmax" equal to 5 minus "Eastmax" and "Westmax" equal to 6. The difference is not significant and therefore the null hypothesis has to be rejected.

	Mean	CI	Std.	Reduction
Run 1	53,2	+/-7,5	32,2	88,3%
Run 2	61,9	+/-7,5	32,4	88,7%
Run 3	52,0	+/-7,4	31,9	88,5%

**Table 4.** Difference with Common Random Numbers

Table 4 shows the results for the same experiment with common random numbers. The variance reduction leads to a great reduction of the confidence interval length. The difference of the mean is significant for all three simulation runs and therefore the null hypothesis cannot be rejected.

**Resumé:** This work shows a new and different approach of solving comparisons. Visual Basic for Applications (VBA) is used to program the class modules of source, queue, server and sink. Furthermore, the event logic and the event handler are directly programmed and no existing tool, software or simulator is used. Therefore the model is created without any graphical aid.

**Corresponding author:** Thomas Löscher,  
Department of Analysis and Scientific Computing  
Vienna University of Technology  
Wiedner Hauptstraße 8-10, 1040 Vienna, Austria  
[thomas@loescher.at](mailto:thomas@loescher.at)

Received: June 20, 2007

Reviewed: September 19, 2007

Accepted: October 10, 2007