
Article:
Möller, J., Davis, S., Stevenson, M. et al. (1 more author) (2017) Validation of a DICE Simulation Against a Discrete Event Simulation Implemented Entirely in Code. Pharmacoeconomics. ISSN 1170-7690

https://doi.org/10.1007/s40273-017-0534-0

Validation of a DICE Simulation against a Discrete Event Simulation Implemented Entirely in Code

Short Running Title: DICE Validation

Authors: Jörgen Möller, MSc[1]; jorgen.moller@evidera.com

Sarah Davis, MPhys[2]; s.davis@sheffield.ac.uk

Matt Stevenson, PhD[2]; m.d.stevenson@sheffield.ac.uk

J. Jaime Caro, MDCM[3,4]; jaime.caro@mcgill.ca

Institutions:

[1] Modeling & Simulation, Evidera, 1 Butterwick, London, W6 8DL, United Kingdom

[2] School of Health and Related Research, University of Sheffield, 30 Regent Street. Sheffield, S1 4DA United Kingdom

[3] Epidemiology & Biostatistics, McGill University, 1020 Pine Avenue W, Montreal, Canada H3A 1A2

[4] Evidera, 500 Totten Pond Road, 5th Floor, Waltham, MA 02451 USA

Corresponding Author:

J. Jaime Caro, 39 Bypass Road, Lincoln MA, 01773 +1 978 760 4627

Jaime.caro@mcgill.ca

Required Information: This work has not been presented at any congresses or meetings.

Word Count: Abstract = 216 words/250 max; Manuscript = 2,323 words (Introduction through Conclusion); Tables = 2; Figures = 2

ABSTRACT

Background. Modeling is an essential tool for health technology assessment, and various techniques for conceptualizing and implementing such models have been described. Recently, a new method has been proposed—the discretely-integrated condition event or DICE simulation—that enables frequently employed approaches to be specified using a common, simple structure that can be entirely contained and executed within widely available spreadsheet software. To assess if a DICE simulation provides equivalent results to an existing discrete event simulation a comparison was undertaken.

Methods. A model of osteoporosis and its management programmed entirely in Visual Basic for Applications and made public by the NICE Decision Support Unit was downloaded and used to guide construction of its DICE version in Microsoft Excel®. The DICE model was then run using the same inputs and settings, and the results were compared.

Results. The DICE version produced results that are nearly identical to the original ones, with differences that would not affect the decision direction of the incremental cost-effectiveness ratios (<1% discrepancy), despite the stochastic nature of the models.

Limitation: The main limitation of the simple DICE version is its slow execution speed.

Conclusions: DICE simulation did not alter the results and, thus, should provide a valid way to design and implement decision-analytic models without requiring specialized software or custom programming. Additional efforts need to be made to speed up execution.

KEY POINTS FOR DECISION MAKERS

- DICE simulation provides a simple common framework for specifying health technology assessment models entirely in a spreadsheet

- DICE simulation replicates a discrete event simulation without requiring any custom software code.

- The use of DICE simulation introduced no bias or alteration of results compared to a traditional implementation

- Execution with the basic DICE macro can be slow.

1 INTRODUCTION

To inform health authorities who make decisions about funding new health care interventions, analysts often use health economic models [1]. These models are mathematical frameworks that relate the course of the illness to the use of the alternative interventions and other factors. By using such a model, the often limited clinical trial data can be extrapolated to longer periods of time, other populations, and different practices than those studied in the trials; moreover, aspects such as the expected costs and impact on quality of life can be incorporated. The models also allow for the quantification of uncertainty around the results. A recent task force proposed good practice guidelines for the conceptualization, construction, validation, and analyses of health economic models [2].

There are several ways of classifying the methodological approaches to structuring a health economic model (see, for example, [3, 4]), but one simple distinction has to do with the methods used to simulate the population's course over time. One option is to define the states that people can be in, and consider how the population distributes among those states at various time points [5]. This cohort method yields deterministic results and requires relatively few calculations, but it does not readily address heterogeneity in the determinants of the course (e.g., age, sex, genetic makeup) [6], or competing risks [7]. Complex pathways (e.g., involving multiple treatment switches) are also cumbersome to implement in a cohort model [8]. The other major option is to simulate the population by separately considering what might happen to each individual explicitly [9]. The structure of a patient-level model, or micro-simulation, may be limited to states the person can be in, or may focus on the events that occur (i.e., unconstrained discrete event simulation, DES [10]). Either way, this individual-sampling approach addresses many of the

shortcomings of the cohort method, but it yields stochastic results and is much more computationally intensive due to the need to run sufficient patients through the model to generate stable estimates [11]. A guidance on the development of patient-level models and their use for health technology assessment has been published by the National Institute for Health and Care Excellence (NICE) [12].

Recently, another alternative for designing and structuring health economic models—discretely-integrated condition event (DICE) simulation—has been described [13]. This approach involves conceptualizing the problem in terms of the aspects that exist over time ("conditions"), those that occur at points in time ("events") and their discrete-integration in terms of the consequences of any one event or condition for the others. It turns out that all of the commonly-used modeling techniques described above are encompassed by DICE. For example, Markov states can be thought of as conditions to which some restrictions are imposed (e.g., mutual exclusivity), and transitions are consequences of a recurring transition event. This model can be implemented in a spreadsheet, such as Microsoft (MS) Excel® by tabulating the lists of conditions and events and specifying the consequences of each event in additional tables. The discrete integration can be accomplished via a simple macro (Figure 1) that loops through the event tables and carries out the specified instructions. These are written as text expressions using appropriate worksheet functions and syntax, but without the equal sign that activates them.

As the DICE simulation approach has been delineated only recently, it was not considered in any of the various earlier modeling guidelines. There is, thus, interest in testing it to ensure that a model structured in this way yields results that are consistent with a more traditional approach. One way to accomplish this validation is to take an existing model

and convert it to its DICE counterpart. The two versions can then be analyzed and various aspects compared, including the computation time. Of particular interest is the degree to which the results from the DICE version accord with those of the original. In this paper, such a validation is reported.

2 METHODS

In 2014, the Decision Support Unit (DSU) of NICE produced a guidance on the development and use of patient-level simulation for health technology assessment (HTA) [12]. As part of that work, several versions were prepared of a model that addressed the cost-effectiveness of various osteoporosis treatments. In particular, that problem was modeled using a DES as well as an individual state-transition approach. Although somewhat simplified for didactic reasons, the problem presented sufficient complexity to illustrate the advantages and disadvantages of these techniques. The various models are clearly and extensively described in the comprehensive technical support document (TSD) [12] and they can be downloaded from http://www.nicedsu.org.uk/Patientlevel-simulation-TSD(2892880).htm.

The DSU efforts provided a unique opportunity to validate DICE simulation. For this purpose, we downloaded the osteoporosis individual models programmed in MS Excel® from the DSU website (http://www.nicedsu.org.uk/TSD15_Excel_code.zip) and reformulated the DES (Excel_DSU_VBA_DES.xlsm) as a DICE model. The DES considered hip and vertebral fractures, and death from either a hip fracture or other causes. Patient attributes were the history of fractures, current utility, and various times to events. The global variables included the costs of fracture and treatment, utility at baseline, utility multipliers post-fracture, the failure-time distributions (which determine time to hip fracture, vertebral fracture, or death), intervention effects, outcomes such as

quality-adjusted life-years (QALYs), life years (LYs) and discounting factors. The model simulates one patient at a time by first initializing the global variables, other than the total costs and QALYs, and then setting that patient's attributes. The next event is determined and processed in terms of its effect on costs, QALYs, and times of remaining events. Patient utility and history are also updated. This continues until the patient dies or the time horizon is reached; then the next patient is simulated. The process is repeated for each intervention (having stored the random number sequences to reduce nuisance variance). The entire model is coded in Visual Basic for Applications (VBA); and this code is specific to this particular model. The MS Excel® worksheets simply provide a place to store and display the results. Other versions in R, SIMUL8®, and TreeAge Pro® were also produced to accompany the NICE TSD.

The DICE version of the model was created using the template DICEd4.xlsm, downloaded from http://www.Evidera.com/DICE. It has six events: apart from Start and End (mandatory events in DICE), there are HipFract, VertebralFrac, DeathHipFract, and Death. As there are no entities or attributes in a DICE, the ongoing information is stored in 22 conditions covering utilities, costs, event times, and random numbers. Twelve accumulator outputs are defined (LYs, QALYs, and four costs; together with their discounted counterparts), as well as four counters (death due to hip fracture, other death, age at death, and replication number). In the Start event, all required conditions and outputs are initialized, and the times of fracture and other death are sampled from their distributions. The model then determines the lowest event time and proceeds to execute the expressions in the corresponding table. The two fracture events have similar structures: both update the fracture history, utility, and age; record the time of the event; and accrue LYs, QALYs, and costs. In the vertebral fracture event, a time to the next vertebral fracture is sampled to allow a second vertebral fracture to occur (after which no

further vertebral fractures are allowed), whereas in the hip fracture event, the probability of death following hip fracture is used to determine whether the hip fracture is fatal. In the Death event, whether from a hip fracture or other causes, only the age at death is recorded and the death is counted. In the End event, regardless of what has happened before, the final accrual of LYs, QALYs, and costs is tallied before the same patient is simulated using the next treatment (Figure 2).

The DICE model is entirely specified in a MS Excel® workbook, with a worksheet for Conditions, one for Events, and another for Outputs. All of the expressions that operate the model are tabulated in the corresponding Event tables. These are executed by a simple macro that loops through each table, row-by-row, inserting sequentially an "=" in front of each expression to convert it into an active MS Excel® function. The macro can be viewed in the VBA module named 'DICEd' of the template referred to above. This same macro will execute any DICE model specified in appropriate MS Excel® tables.

Once the DICE version of the model was created, it was run with the same inputs as the DSU used in its analyses. A hypothetical treatment was compared to no intervention in 50,000 patients. Apart from examining the results, the running times were also compared. This was done once using the parameters from the original DSU model, and once using a different set of input parameters to ensure that the findings were not dependent on the exact specification of the original DSU model. This alternative parameterization is reported in more detail as it included a cost for the 'no treatment' arm, making it possible to check that treatment costs were being appropriately handled in both treatment arms. The parameter sets for each scenario are given in Table 1.

3 RESULTS

The DES implemented entirely in VBA by the DSU occupied 233 lines of code. The equivalent generic DICE macro is 70 lines of code. It took 11 hours to carry out the conversion. This involved understanding the design of the model and restructuring it into events, conditions, and outputs. For 50,000 patients and two treatment strategies (osteoporosis treatment and no treatment), the DES in VBA takes 1.64 seconds; for an equivalent number of patients and treatment strategies, the basic DICE version takes 33 minutes.

For the original DSU parameterization, the absolute costs and QALY for each arm in DICE were within 1% of the values obtained by the DSU DES. Having confirmed that the results were similar for this scenario, the results were then compared for the alternative parameterization and also found to be within 1% of the values obtained by the DSU DES except for the undiscounted lLYs gained and the number of vertebral fractures prevented. The difference between the two models in undiscounted LYs was slightly more than 5% for each arm, but the incremental LYs are small, making any difference between the models appear greater in percentage terms. The absolute difference was less than 1.5 days.

4 DISCUSSION

The results of the analyses carried out demonstrate that the DICE implementation yields equivalent results. In other words, formulating a model using the DICE specification does not alter the outcomes. This is important because DICE offers some distinct advantages. As the entire model is specified completely in the Excel tables, it is very transparent. Nothing specific to the model is in code and a reviewer does not need to learn new

software—understanding spreadsheets and the way formulas are writtern in Excel is sufficient. By the same token, the macro that runs a DICE is unchanged from model to model: no programming is required, it need not be reverified and a user or reviewer does not need to re-examine it. DICE models can be created very quickly and modifications are a simple matter of inserting or deleting rows or editing the text expressions, with no need to reconnect formulas or rewrite code. This makes structural uncertainty analyses relatively simple. The DICE specification is also very flexible, allowing combination of state-transition and time-to-event components in a single model, thus allowing a modeler to leverage the best features of each approach as appropriate.

The most obvious limitation of the DICE approach, which was apparent during this validation exercise, was the extended model run-time. This occurs in the basic DICE because of the interaction between the VBA code and the spreadsheet formulae. Each time this happens, Excel triggers the worksheets, slowing down execution. This can be much reduced by reading the Conditions and Event tables into memory and executing calculations there, thus minimizing the number of times the macro interacts with the worksheets. This preserves the generic nature of the macro and the transparent specification of the models but can cut runtime substantially (to 14.3 minutes for the DSU model). A version of DICE that can achieve this (EviDICE) has been made freely available to HTA agencies and academic groups for non-commercial use. A faster version that uses a compiled macro should be available shortly.

One of the key advantages of implementing a model in DICE is that it provides a single template for implementing a variety of model structures. Once the modeling community has built up sufficient familiarity with the DICE framework, this should make models easier to develop and validate, as users will know where in the model to look to find

particular information and model functionality. Various courses and workshops at major meetings have been held to increase familiarity with the methods.The availability of a standardized modelling framework which is freely available may also encourage a broader group of modelers to use a time-to-event (DES) modelling structure; previously many modelers have avoided using DES due to a lack of affordable bespoke simulation software or insufficient programming skills to implement one confidently in VBA or R [14]. Therefore, the availability of DICE may result in improved models in situations where a model could be built using a state-transition approach but a DES approach is more parsimonious. One example of such a situation is when the Markov assumption does not hold, but other situations are described in the DSU's TSD on patient-level simulation [12]. Regardless of the model structure implemented, the developer needs to understand the DICE framework and correctly convert their conceptual model into a set of conditions and events with correct expressions to update the conditions dependent on the events.

Furthermore, the implementation of the model using MS Excel® formulae typed as text without equal signs may make validation and de-bugging more difficult. Whilst the results of the formulae inputted as text in a single cell can be checked by simply inserting the equal sign, this gives the value only under current conditions. In order to step through the model for de-bugging purposes, it is necessary to turn on the logging function which outputs to a text file every execution step taken by the macro along with changes to Condition levels, Event times or Outputs. This is a bit more complex than stepping through a model coded entirely in VBA, where tools such as the watch window or the locals window can be used to track the impact of each line of code on variables that are held within the VBA.

A limitation of the validation of DICE reported here is that whilst the model was built by one author [JM] and validated by a second author [SD], the validation did not include an exhaustive examination of the formulae and VBA code in the DICE model. Instead we relied on comparing results against a previously validated implementation of the DSU DES model coded using VBA. However, the fact that the results compare well when using both the original and an alternative parameter set is strongly supportive of the DICE and VBA model implementations being equivalent.

5 CONCLUSION

DICE simulation offers a means to design and implement a decision-analytic model without having to resort to specialized software or to engage in custom programming. The results are not distorted by the implementation and the formulation is very transparent as the entire model is specified in simple tables. Speed of execution, however, remains a concern. Various educational initiatives are underway to help familiarize modelers with the method.

6 DATA AVAILABILITY STATEMENT

The original DSU model and the DICE version can be downloaded from the DSU website (http://www.nicedsu.org.uk/TSD15_Excel_code.zip) while the template used to build the DICE version can be downloaded from htttp://www.evidera.com/DICE.

7 ACKNOWLEDGEMENTS

No one else contributed to this work.

8 COMPLIANCE WITH ETHICAL STANDARDS

8.1 Funding

No financial support was provided for this work. JM and JJC continued to receive their salaries from Evidera during the time they developed the concepts in this paper. However, the agreement with their employer ensured their independence in study design, data interpretation, writing the manuscript, and publishing the report. SD and MDS have no conflicts of interest to report.

8.2 Conflict of Interest Statements

Jörgen Möller and J.Jaime Caro continued to receive their salaries from Evidera during the time they developed the concepts in this paper. However, the agreement with their employer ensured their independence in study design, data interpretation, writing the manuscript, and publishing the report. Sarah Davis and Matt Stevenson have no conflicts of interest to report.

8.3 Contributions Made by Each Author

All authors contributed to the text, reviewed the final version and approved it. JM created the DICE version, SD and MS created the original DSU version and ran the comparisons with the DICE version.

8.4 Overall Guarantor

J.Jaime Caro will serve as overall guarantor for this study and manuscript.

8.5 Ethical Approvals and Informed Consent

For this type of study, formal consent is not required, nor is any informed consent process. This article does not contain information from any studies with human participants or animals performed by any of the authors.

9 REFERENCES

1. Caro JJ. Disease-simulation models and health care decisions. CMAJ. 2000;162(7):1001-2.

2. Caro JJ, Briggs AH, Siebert U, Kuntz KM, Force I-SMGRPT. Modeling good research practices--overview: a report of the ISPOR-SMDM Modeling Good Research Practices Task Force--1. Value Health. 2012;15(6):796-803. doi:10.1016/j.jval.2012.06.012.

3. Brennan A, Chick SE, Davies R. A taxonomy of model structures for economic evaluation of health technologies. Health Econ. 2006;15(12):1295-310. doi:10.1002/hec.1148.

4. Stahl JE. Modelling methods for pharmacoeconomics and health technology assessment: an overview and guide. Pharmacoeconomics. 2008;26(2):131-48.

5. Siebert U, Alagoz O, Bayoumi AM, Jahn B, Owens DK, Cohen DJ et al. State-transition modeling: a report of the ISPOR-SMDM Modeling Good Research Practices Task Force--3. Value Health. 2012;15(6):812-20. doi:10.1016/j.jval.2012.06.014.

6. Zaric GS. The impact of ignoring population heterogeneity when Markov models are used in cost-effectiveness analysis. Med Decis Making. 2003;23(5):379-96. doi:10.1177/0272989X03256883.

7. Wolkewitz M, Cooper BS, Bonten MJ, Barnett AG, Schumacher M. Interpreting and comparing risks in the presence of competing events. BMJ. 2014;349:g5060. doi:10.1136/bmj.g5060.

8. Tosh J, Stevenson M, Akehurst R. Health economic modelling of treatment sequences for rheumatoid arthritis: a systematic review. Curr Rheumatol Rep. 2014;16(10):447. doi:10.1007/s11926-014-0447-2.

9. Stevenson MD, Brazier JE, Calvert NW, Lloyd-Jones M, Oakley JE, Kanis JA. Description of an individual patient methodology for calculating the cost-effectiveness of treatments for osteoporosis in women. J Oper Res Soc. 2005;56(2):214-21. doi:10.1057/palgrave.jors.2601903.

10. Caro JJ, Möller J, Karnon J, Stahl J, Ishak J. Discrete Event Simulation for Health Technology Assessment. Boca Raton, FL: Chapman and Hall/CRC Press; 2015.

11. Griffin S, Claxton K, Hawkins N, Sculpher M. Probabilistic analysis and computationally expensive models: Necessary and required? Value Health. 2006;9(4):244-52. doi:10.1111/j.1524-4733.2006.00107.x.

12. Davis S, Stevenson M, Tappenden P, Wailoo AJ. NICE Decision Support Unit (DSU) Technical Support Document 15: Cost-effectiveness modelling using patient-level simulation. Report by the Decision Support Unit. National Institute for Health and Care Excellence (NICE), London. 2014. http://www.nicedsu.org.uk/Patientlevel-simulation-TSD(2892880).htm. Accessed June 2016.

13. Caro JJ. Discretely Integrated Condition Event (DICE) Simulation for Pharmacoeconomics. Pharmacoeconomics. 2016;34(7):665-72. doi:10.1007/s40273-016-0394-z.

14. Stevenson MD, Simpson EL, Rawdin AD, Papaioannou DE. A review of discrete event simulation in National Coordinating Centre for Health Technology Assessment-funded work and a case study exploring the cost-effectiveness of testing for

thrombophilia in patients presenting with an initial idiopathic venous thromboembolism. Journal of Simulation. 2010;4(1):14-23. doi:10.1057/jos.2009.12.

10 TABLES

Table 1. Parameter values for original DSU model and alternative parameterization

Parameter description	Value in original DSU model	Value in alternative parameterization
Weibull curve for time to hip fracture:		
Shape (alpha)	4	3.5
Scale (beta)	10	8
Weibull curve for time to 1st vertebral fracture:		
Shape (alpha)	2	2.5
Scale (beta)	8	7
Weibull curve for time to 2nd vertebral fracture		
Shape (alpha)	2	2.5
Scale (beta)	8	7
Normal distribution for time to death from other causes		
Mean (mu)	12	10
SD (sigma)	3	2.5
Hip fracture mortality probability	0.05	0.02
Acceleration factors		
Hip fracture	2	2.5
First vertebral fracture	2	1.5
Baseline utility	0.70	0.90
Utility decrements (multipliers):		
Post-hip fracture	0.75	0.60
Post-vertebral fracture	0.90	0.95
Post-vertebral fracture 2	1.00	1.00

Parameter description	Value in original DSU model	Value in alternative parameterization
Cost of osteoporosis treatment (per year)	£500	£5000
Cost of no intervention (per year)	0	£100
Cost of a hip fracture (per event)	£7000	£6000
Cost of a vertebral fracture (per event)	£3000	£1500

Abbreviations: DSU: Decision Support Unit; SD: Standard deviation

Table 2. Results of the simulation comparing the DICE version with the DES in VBA, for alternative parameter inputs

Incremental outcomes*	DES in VBA	DICE version	Difference (DICE vs. DES in VBA)
Vertebral fractures prevented (per 1000 patients)	421	416	-1.1%
Hip fractures prevented (per 1000 patients)	691	693	0.3%
Hip fracture deaths prevented (per 1000 patients)	13.9	13.8	-0.6%
Life years (undiscounted)	0.059	0.063	5.9%
QALYs	0.836	0.841	0.6%
Treatment cost	£41,018	£41,146	0.3%
Total cost (lifetime)	£37,139	£37,259	0.3%
ICER (cost/QALY undiscounted)	£39,548	£39,429	-0.3%
ICER (cost/QALY discounted)	£44,403	£44,280	-0.3%
ICER (cost/LYG undiscounted)	£744,370	£705,366	-5.2%

* Incremental outcomes for treatment versus no intervention with costs and QALYs discounted at 3.5% unless otherwise stated.

Abbreviations: DES: Discrete event simulation; DICE: Discretely-integrated condition event; ICER: Incremental cost-effectiveness ratio; QALY: Quality-adjusted life-year; VBA: Visual Basic for Applications.

11 FIGURE LEGENDS

Figure 1. Loop that implements the discrete integration in VBA (Visual Basic for Applications)

Figure 2. Schematic of the DICE version of the osteoporosis model

12 FIGURES

Figure 1. Loop that implements the discrete integration in VBA (Visual Basic for Applications)

```
Do
    ThisEventName = UCase(EventTypes(Range("NextEvent"), 3))
    ThisEvent = Range(EventTypes(Range("NextEvent"), 3) & "Rge")
    For i = 1 To UBound(ThisEvent, 1)
        If ThisEvent(i, 3) <> "" Then
            Range("CalcCell") = "=" & ThisEvent(i, 3)
            Range(ThisEvent(i, 2)) = Range("CalcCell").Value
        End If
    Next i
    Range("CalcCell") = ""      'Housekeeping
Loop Until ThisEventName = "END"
```

Figure 2. Schematic of the DICE version of the osteoporosis model