Appendix A: Visual basic code for discrete event simulation example model
The text below can be pasted into an EXCEL Visual Basic Module. Doing so will display the code specific formatting which makes the code more readable. If the code is to be run successfully, the module should be within a workbook containing one worksheet named ‘results’ and another named ‘PL_results’. On the ‘results’ sheet the following names should be given to the cells where the comparator arm results are to be pasted; ResCosts, ResQALYs, ResDCosts, ResDQalys. The cells to the right of these should be left free for the intervention arm results. In the ‘PL_results’ sheet, cells A1:L1 should be named ‘pl_results_header’. There should also be one cell named ‘timetaken’ on the results sheet. An example Excel workbook containing both the Visual Basic Module and formatted results sheets can be downloaded from the DSU website.

Before running the code, set the number of patients required by defining the ‘nPatients’ variable and set the variable ‘dbg’ to TRUE for patient level results or FALSE for cohort level results. [see the text indicated by ‘<<<<<<<<<<<<<<<’ below]

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Option Explicit

Dim nPatients As Long ' size of cohort [nb using 'dim as long' rather than 'dim as integer' for integer variables. This is particularly important for integers that may go above 30000 as this prevents overflow errors]

Dim horizon As Double ' time horizon in years (must be less than largenumb variable)

Dim baseU As Double ' baseline utility

Dim hipWeibAlpha As Double ' alpha parameter for Weibul used to sample time to hip fracture

Dim hipWeibBeta As Double ' beta parameter for Weibul used to sample time to hip fracture

Dim vertWeibAlpha As Double ' alpha parameter for Weibul used to sample time to vertebral fracture

Dim vertWeibBeta As Double ' beta parameter for Weibul used to sample time to vertebral fracture

Dim meanSurvival As Double ' mean survival used in normal distribution for all cause mortality
Dim SDsurvival As Double ' standard deviation for survival used in normal distribution for all cause mortality

Dim TThipfrac As Double ' time to hip fracture from simulation start

Dim TTvertfrac As Double ' time to first vertebral fracture from simulation start

Dim TTvertfrac2 As Double ' time from first vertebral fracture to second vertebral fracture

Dim TTdeath As Double ' time to death from simulation start

Dim TTlastevent As Double ' time of most recent event

Dim TTnextEvent As Double ' keeps track of time of next event

Dim hipfracmort As Double ' probability of death occurring at time of hip fracture (after costs of hip fracture are incurred)

Dim vertcost As Double ' cost of vertebral fracture

Dim hipcost As Double ' cost of hip fracture

Dim vertUmult As Double ' utility multiplier applied after first vertebral fracture

Dim hipUmult As Double ' utility multiplier applied after first hip fracture

Dim hipstatus As Long ' zero if no previous hip fracture, and 1 after first hip fracture

Dim vertstatus As Long ' zero if no previous hip fracture, 1 after 1st vertebral fracture, 2 after 2nd

Dim Hdeathstatus As Long ' zero if no death due to hip fracture

Dim ACdeathstatus As Long ' zero if no death due to non-hip fracture cause

Dim CostsAccrued As Double ' patient costs accrued so far in simulation

Dim QALYsAccrued As Double ' patient QALYs accrued so far in simulation
Dim TCostsAccrued As Double ' total costs accrued so far across patient cohort
Dim TQALYsAccrued As Double ' total QALYs accrued so far across patient cohort
Dim hipfracCount As Long ' tally of hip fractures
Dim vertfracCount As Long ' tally of vertebral fracture
Dim hipdeathcount As Long ' tally of deaths from hip fractures
Dim deathallcause As Long ' tally of deaths from all causes (other than hip fracture)
Dim Cost_Inter_pa As Double ' cost of intervention per annum
Dim hipEfficacy As Double ' hip efficacy variable, applied as a multiplier to the sampled time to event for control
Dim vertEfficacy As Double ' vert efficacy variable, applied as a multiplier to the sampled time to event for control
Dim DiscC As Double ' discounting rate for costs
Dim DiscQ As Double ' discounting rate for QALYs
Dim DiscCi As Double ' instantaneous discount rate for costs
Dim DiscQi As Double ' instantaneous discount rate for QALYs
Dim DCostsAccrued As Double ' Discounted costs accrued so far for patient
Dim DQALYsAccrued As Double ' Discounted QALYs accrued so far for patient
Dim DiscTCostsAccrued As Double ' Total discounted costs accrued so far across patient cohort
Dim DiscTQALYsAccrued As Double ' Total discounted QALYs accrued so far across patient cohort
Dim rand() As Double ' array of random numbers used to sample time to: 1. hip fracture, 2. vertebral fracture, 3. 2nd vertebral fracture, 4. death (all cause),
‘5. hip fracture mortality

Dim j As Long ' Long used to fill random number array
Dim k As Long ' Long used loop through filling multiple random number arrays
Dim i As Long ' integer used to loop through treatments
Dim n As Long ' integer used to loop through patients in cohort
Dim resultsSheet As Worksheet ' name used to refer to worksheet where results are pasted
Dim router As Long ' used to select actions according to next event
Dim currentU As Double ' keeps track of current utility
Dim largenumb As Double ' used to force end of simulation when death occurs
Dim PL_output() As Double ' used to output patient level results when debug code is switched on
Dim dbg As Boolean ' used to switch on debugging code
Dim starttime As Double ' used to record run time
Dim endtime As Double ' used to record run time

Sub DES_osteo_example()}
starttime = Timer

nPatiens = 50000 ' SET NUMBER OF PATIENTS HERE <<<<<<<<<<<<<<<<

ReDim PL_output(nPatients - 1, 11)
'columns of this array are 0:patient number, 1:TThipfrac, 2:TTvertfrac, 3:TTvertfrac2, 4:TTdeath, 5:hipcount, 6:vertcount, 7:hipdeathcount,
'8: deathallcause, 9:DCostsAccrued, 10:DQALYsAccrued, 11:Discounted treatment costs

horizon = 30
largenum = 10000000

ReDim rand(1 To 5, 0 To nPatients - 1) ' defines arrays of random numbers. Array size set to cohort size, but this assumes each used once per patient
    'so increase array size if repeated use occurs
dbg = True ' set to TRUE to output patient level results HERE <<<<<<<<

If dbg Then
    Sheets("PL_results").UsedRange.Offset(1).ClearContents 'clear old patient-level data
    Sheets("PL_results").UsedRange 'Excel needs nudging to reset the used range property of a worksheet
End If
For k = 1 To 5
    Rnd -2
    Randomize
    Choose(k, 11, 3, 5, 13, 17) ' this lines sets different seeds for each random number used to estimate different quantities e.g hip fracture, vert fracture
For j = 0 To nPatients - 1 ' this loop fills the random number sequences so they are the same from treatment to control and
     'the same stream from one excecution to the next
        rand(k, j) = Rnd(1)
    Next j
Next k

'set global variables that are fixed across all patients and both treatment strategies
baseU = 0.7
hipfracmort = 0.05
hipcost = 7000
vertcost = 3000
vertUmult = 0.9
hipUmult = 0.75
DiscC = 0.035
DiscQ = 0.035
DiscCi = Log(1 + DiscC)
DiscQi = Log(1 + DiscQ)
hipWeibAlpha = 4
hipWeibBeta = 10
vertWeibAlpha = 2
vertWeibBeta = 8
meanSurvival = 12
SDsurvival = 3

i = 0

For i = 0 To 1 ' this FOR....NEXT statement repeats the model for each treatment strategy
these statements clear the values ready for the new treatment

TCostsAccrued = 0
TQALYsAccrued = 0
DiscTCostsAccrued = 0
DiscTQALYsAccrued = 0
hipfracCount = 0
vertfracCount = 0
hipdeathcount = 0
deathallcause = 0

these statements set intervention costs and efficacy data for treatment (i=1) and control (i=0)

If i = 0 Then
    Cost_Inter_pa = 0 ' drug cost for comparator arm
    hipEfficacy = 1 ' hip efficacy for comparator arm
    vertEfficacy = 1 ' vert efficacy for comparator arm
Else
    Cost_Inter_pa = 500 ' drug cost for intervention arm
hipEfficacy = 2 ' hip efficacy for intervention arm

vertEfficacy = 2 ' vert efficacy for intervention arm

End If

n = 0 ' n is used as index to loop through patients

For n = 0 To nPatients - 1 ' this FOR...NEXT statement repeats the simulation until whole cohort has been simulated

'[condition applies from here until 'Next n' statement below]

'variables set for this patient

'sample time to event data for this patient

TThipfrac = hipEfficacy * hipWeibBeta * (-Log(rand(1, n))) ^ (1 / hipWeibAlpha) 'sampling from weibull in excel: beta * (-ln(rand()))^(1/alpha)

TTvertfrac = vertEfficacy * vertWeibBeta * (-Log(rand(2, n))) ^ (1 / vertWeibAlpha)

TTvertfrac2 = vertWeibBeta * (-Log(rand(3, n))) ^ (1 / vertWeibAlpha) ' note efficacy not applied to second vert frac in simul8 model

TTdeath = Application.WorksheetFunction.NormInv(rand(4, n), meanSurvival, SDsurvival) ' sampling from a normal distribution
TTdeath = WorksheetFunction.Max(TTdeath, 0) 'This just eliminates negative time to death samples which are unlikely but possible given the normal distribution used.

If dbg Then

PL_output(n, 0) = n
PL_output(n, 1) = TThipfrac
PL_output(n, 2) = TTvertfrac
PL_output(n, 3) = TTvertfrac + TTvertfrac2
PL_output(n, 4) = TTdeath
End If

'initialise patient starting variables and wipe clean any values carried over from previous patient

TTlastevent = 0
currentU = baseU
hipstatus = 0
vertstatus = 0
CostsAccrued = 0
QALYsAccrued = 0
DCostsAccrued = 0
DQALYsAccrued = 0
Hdeathstatus = 0
ACdeathstatus = 0

Do While TTlastevent < largenum
  'this DO WHILE....LOOP statement allows events to be processed one by one until
  'death occurs or time horizon is reached as indicated by TTlastevent=largenum

  'this block determines order of events, sets the router variable to select that event and sets
  ' TTnextEvent variable to time of that event

  TTnextEvent = TTvertfrac
  router = 1

  If TThipfrac < TTnextEvent Then
    router = 2
    TTnextEvent = TThipfrac
End If

If TTdeath < TTnextEvent Then
    router = 3
    TTnextEvent = TTdeath
End If

'end of block checking order of events

If TTnextEvent < horizon Then 'this IF..THEN..ELSE..END IF statement stops the patient experiencing the
    'next event if it occurs after the end of the simulation for this patient
'NOTE it isn't necessary to specify a time horizon for a DES as all patients could be followed until death, but
'this is included here to allow comparison against MARKOV model structures
Select Case router ' this SELECT CASE statement processes the next event outcomes according to the router variable

Case 1 'processes vertebral fractures

CostsAccrued = CostsAccrued + vertcost

' applies cost of vertebral fracture

DCostsAccrued = DCostsAccrued + vertcost / ((1 + DiscC) ^ TTvertfrac)

'discounted cost of vert frac (one-off cost so discount factor at time of event is applied)

QALYsAccrued = QALYsAccrued + ((TTvertfrac - TTlastevent) * currentU)

' accrues QALYs by applying the utility value applies before this event to the time between the last event and this one

DQALYsAccrued = DQALYsAccrued + currentU * (Exp(TTvertfrac * (0 - DiscQi)) - Exp(TTlastevent * (0 - DiscQi))) / (0 - DiscQi)

' accrues discounted QALYs

TTlastevent = TTvertfrac ' updates time of most recent event to time of this event

If vertstatus = 0 Then ' the following block occurs if this event is the first vertebral fracture

currentU = currentU * vertUmult ' updates current utility to the value that applies from this event to next

TTvertfrac = TTvertfrac + TTvertfrac2 ' updates time of next vertebral fracture allowing second vertebral fractures to occur

vertstatus = 1 ' records that one vertebral fracture has occurred
Else 'the following block occurs if this event eis the second vertebral fracture

\[ \text{currentU} = \text{currentU} \] ' NB: no utility decrement for second vert fracture

\[ \text{TTvertfrac} = \text{largenum} \] ' stops any subsequent vertebral fractures occuring (i.e max of 2 per patient)

\[ \text{vertstatus} = 2 \]

End If

Case 2 ' processes hip fracture

\[ \text{CostsAccrued} = \text{CostsAccrued} + \text{hipcost} \] ' accrues cost of fracture

\[ \text{QALYsAccrued} = \text{QALYsAccrued} + ((\text{TThipfrac} - \text{TTlastevent}) \times \text{currentU}) \] ' accrues QALYs from last event to this event

\[ \text{DCostsAccrued} = \text{DCostsAccrued} + \text{hipcost} / ((1 + \text{DiscC}) \times \text{TThipfrac}) \] ' accrues discounted cost (one-off cost so discount factor at time of event applied)

\[ \text{DQALYsAccrued} = \text{DQALYsAccrued} + \text{currentU} \times (\exp(\text{TThipfrac} \times (0 - \text{DiscQi})) - \exp(\text{TTlastevent} \times (0 - \text{DiscQi}))) / (0 - \text{DiscQi}) \]
' accrues discounted QALYs

currentU = currentU * hipUmult ' updates utility

TTlastevent = TThipfrac ' records this time as time of last event

If rand(5, n) < hipfracmort Then ' checks to see if hip fracture results in death

  CostsAccrued = CostsAccrued + TThipfrac * Cost_Inter_pa ' treatment costs accrued from start till death
  DCostsAccrued = DCostsAccrued + Cost_Inter_pa * (Exp(TThipfrac * (0 - DiscCi)) - Exp(0 * (0 - DiscCi))) / (0 - DiscCi)

  ' discounted treatment costs (cost per annum is multiplied by discounted years)

  If dbg Then
    PL_output(n, 11) = Cost_Inter_pa * (Exp(TThipfrac * (0 - DiscCi)) - Exp(0 * (0 - DiscCi))) / (0 - DiscCi)
  End If

  TTlastevent = largenum ' forces end of simulation for this patient as death has occurred

  Hdeathstatus = 1

  hipdeathcount = hipdeathcount + 1 ' records fatal hip fracture
End If

TThipfrac = largenumb ' stops subsequent hip fractures

hipstatus = 1 ' records that hip fracture has occurred

Case 3

QALYsAccrued = QALYsAccrued + ((TTdeath - TTlastevent) * currentU) ' accures QALYs from last event to time of death

CostsAccrued = CostsAccrued + TTdeath * Cost_Inter_pa ' accrues costs of treatment from start to time of death

DQALYsAccrued = DQALYsAccrued + currentU * (Exp(TTdeath * (0 - DiscQi)) - Exp(TTlastevent * (0 - DiscQi))) / (0 - DiscQi)

' discounted QALYs

DCostsAccrued = DCostsAccrued + Cost_Inter_pa * (Exp(TTdeath * (0 - DiscCi)) - Exp(0 * (0 - DiscCi))) / (0 - DiscCi)

' discounted cost of treatment (cost per annum is multiplied by discounted years)

ACdeathstatus = 1

If dbg Then

PL_output(n, 11) = Cost_Inter_pa * (Exp(TTdeath * (0 - DiscCi)) - Exp(0 * (0 - DiscCi))) / (0 - DiscCi)

End If
TTlastevent = largenumb 'forces end of simulation for this patient as death has occurred

deadthallcause = deathallcause + 1 ' records non-hip mortality (all cause)

End Select

Else  ' these statements are applied when next event > time horizon

QALYsAccrued = QALYsAccrued + ((horizon - TTlastevent) * currentU)

' this accrues QALYs from last event to end of simulation if patients don't die before horizon

DQALYsAccrued = DQALYsAccrued + currentU * (Exp(horizon * (0 - DiscQi)) - Exp(TTlastevent * (0 - DiscQi))) / (0 - DiscQi)

' discounted QALYs (utility multiplied by discounted years)

CostsAccrued = CostsAccrued + horizon * Cost_Inter_pa ' accrues cost of intervention from start till death

DCostsAccrued = DCostsAccrued + Cost_Inter_pa * (Exp(horizon * (0 - DiscCi)) - Exp(0 * (0 - DiscCi))) / (0 - DiscCi)

' discounted cost of intervention (cost per annum multiplied by discounted years)

TTlastevent = largenumb ' ends simulation if horizon reached

If dbg Then
\[
\text{PL\_output}(n, 11) = \text{Cost\_Inter\_pa} \times (\text{Exp(horizon} \times (0 - \text{DiscCi})) - \text{Exp}(0 \times (0 - \text{DiscCi}))) / (0 - \text{DiscCi})
\]

End If

End If

Loop ' end of DO WHILE loop which stops when TTtastevent = largenumb, i.e death occurred or time horizon reached

TCostsAccrued = CostsAccrued + TCostsAccrued ' adds costs for this patient to total costs for cohort so far

TQALYsAccrued = QALYsAccrued + TQALYsAccrued ' adds QALYs for this patient to total QALYs for cohort so far

DiscTCostsAccrued = DCostsAccrued + DiscTCostsAccrued

DiscTQALYsAccrued = DQALYsAccrued + DiscTQALYsAccrued

hipfracCount = hipfracCount + hipstatus

vertfracCount = vertfracCount + vertstatus
If dbg Then

PL_output(n, 5) = hipstatus
PL_output(n, 6) = vertstatus
PL_output(n, 7) = Hdeathstatus
PL_output(n, 8) = ACdeathstatus
PL_output(n, 9) = DCostsAccrued
PL_output(n, 10) = DQALYsAccrued

End If

Next n ' end of FOR...NEXT loop for n patients

TCostsAccrued = TCostsAccrued / nPatients ' works out average value per patient across cohort

TQALYsAccrued = TQALYsAccrued / nPatients ' works out average value per patient across cohort

DiscTCostsAccrued = DiscTCostsAccrued / nPatients ' works out average value per patient across cohort
DiscTQALYsAccrued = DiscTQALYsAccrued / nPatients ' works out average value per patient across cohort

[ResCosts].Offset(0, i) = TCostsAccrued  'output undiscounted costs
[ResQALYs].Offset(0, i) = TQALYsAccrued  'outputs undiscounted QALYs
[ResDCosts].Offset(0, i) = DiscTCostsAccrued  'outputs discounted costs
[ResDQALYs].Offset(0, i) = DiscTQALYsAccrued  'outputs discounted QALYs

If dbg Then [pl_results_header].Offset(1, 13 * (i)).Resize(nPatients) = PL_output 'output new patient-level data, if desired

' increments i to go to next treatment strategy

Next i  'i loop for looping through treatment strategies

dertime = Timer
Range("timetaken").Value = endtime - starttime
End Sub