Contract No:

This document was prepared in conjunction with work accomplished under Contract No. DE-AC09-08SR22470 with the U.S. Department of Energy (DOE) Office of Environmental Management (EM).

Disclaimer:

SRNL-STI-2017-00729

Date: December 12, 2017

TO: B. T. Butcher, 773-42A

FROM: J. A. Dyer, 773-42A
G. P. Flach, 773-42A

REVIEWER: T. L. Danielson, 703-41A

Topic 3.5.1: Vadose Zone Models

Recommendation 150: Decide how to represent intact and subsided conditions for the proposed new conceptual closure cap design for the purpose of calculating infiltration. Produce new intact and subsided infiltration cases based on new conceptual design.

E-Area LLWF Vadose Zone Model: Probabilistic Model for Estimating Subsided-Area Infiltration Rates

Scope

A probabilistic model employing a Monte Carlo sampling technique was developed in Python to generate statistical distributions of the upslope-intact-area to subsided-area ratio ($Area_{UAi}/Area_{SAi}$) for closure cap subsidence scenarios that differ in assumed percent subsidence and the total number of intact plus subsided compartments. The plan is to use this model as a component in the probabilistic system model for the E-Area Performance Assessment (PA), contributing uncertainty in infiltration estimates (Flach, 2017). Mean values for $Area_{UAi}/Area_{SAi}$ are utilized in Equation (1) to calculate the mean run-on from upslope to an assumed subsided region of the closure cap:

$$\text{Run-on} = (Area_{UAi}/Area_{SAi})(\text{Lateral Drainage} + \text{Surface Run-off})_{\text{HELP Intact Case}} \quad (1)$$

The mean run-on is a key input parameter in the Hydrologic Evaluation of Landfill Performance (HELP) model for generating infiltration rate vs. time profiles for closure cap scenarios involving subsidence. This work was completed in preparation for the next revision of the E-Area Low-Level Waste Facility (LLWF) PA.

Conclusions

Mean values for $Area_{UAi}/Area_{SAi}$ are independent of slope length and increase as percent subsidence decreases toward zero because of the increasing likelihood of long distances of intact area upslope of the subsided region. In addition, mean values for $Area_{UAi}/Area_{SAi}$ increase as the total number of randomly spaced intact and subsided compartments assumed in the probabilistic model increases, reaching a maximum (or conservative upper bound) for an infinite number of randomly spaced compartments. Table 1 summarizes the conservative upper bound values for $Area_{UAi}/Area_{SAi}$ as a function of percent subsidence, which are simply given by:

$$(\text{Area}_{\text{UAi}}/\text{Area}_{\text{Sai}})_{\text{upper bound}} = (100 - \%\ \text{subsidence}) / \%\ \text{subsidence} \quad (2)$$

For example, the value of $(\text{Area}_{\text{UAi}}/\text{Area}_{\text{Sai}})_{\text{upper bound}}$ equals (100 – 10)/10 or 9.0 for the case of 10% subsidence.

Interestingly, the ratios listed in Table 1 are not unique to the "randomly spaced, infinite-hole" case. They also apply to a single subsided compartment located at the toe of the slope, and to multiple subsided compartments when one of the subsided compartments is located at the toe of the slope. The common attribute among these scenarios is a hole at the toe of the slope, which ensures that all runoff from intact portions of the cover system is captured as run-on by subsided portions. Otherwise, runoff from an intact segment below the last hole is lost and average infiltration decreases. While random spacing of a finite number of holes does not ensure a hole at the absolute bottom of the slope, hole density is infinite when the number of holes is infinite, which does ensure a hole at the toe and explains the conservative nature of the infinite-hole case.

For non-bounding cases where a smaller, realistic number of randomly spaced subsided and intact compartments is assumed (e.g., 1 to 6 randomly spaced subsided compartments out of 10 to 100 total compartments), the probabilistic model calculates a mean value for $\text{Area}_{\text{UAi}}/\text{Area}_{\text{SAi}}$ that is less than the corresponding conservative upper bound value listed in Table 1. Highlighted cells and bold entries in this table are further explained in the section *Practical Considerations*.

Table 1. Upslope-Intact-Area to Subsided-Area Ratios for Conservative Upper Bound Cases

Percent Subsidence	**Upslope-Intact-Area to Subsided-Area Ratio for Conservative Upper Bound Case**
1	99
1.6	61.5
2	49
4	**24**
5	19
6.67	**14**
10	9
20	4
100	0

Discussion

Probabilistic Model Source Code and Example Files

The Python source code for the probabilistic subsidence model is included in Appendix A. The user specifies the total number of compartments, average number of subsided compartments, and number of realizations (typically 1,000 to 10,000). Every compartment has a random chance of being subsided during each realization, and the number of subsided compartments per realization (representing one of many slices down the slope width) can range anywhere from zero to the total number of compartments. The constraint, however, is that the average number of subsided compartments must agree with the user-specified input. An example multiple-case batch run (input) file is shown in Figure 1, where each row represents a separate case. The rose-highlighted region in Appendix A defines each of the ten input arguments. The model calculates a mean and median upslope-intact-area to subsided-area ratio for each case.

runPythonFinal.bat

```
SubsidedAverage_Final.py 30 1 10000 table.tab Output_1.txt Summary.txt False w
SubsidedAverage_Final.py 60 3 10000 table.tab Output_2.txt Summary.txt False a
SubsidedAverage_Final.py 60 6 10000 table.tab Output_3.txt Summary.txt False a
SubsidedAverage_Final.py 60 2 10000 table.tab Output_4.txt Summary.txt False a
SubsidedAverage_Final.py 60 3 10000 table.tab Output_5.txt Summary.txt False a
SubsidedAverage_Final.py 60 6 10000 table.tab Output_6.txt Summary.txt False a
SubsidedAverage_Final.py 600 60 1000 table.tab Output_7.txt Summary.txt False a
```

Figure 1. Example Batch Run Input File for Python Probabilistic Subsided-Area Infiltration Model

In addition, the probabilistic model calls a user-added lookup table containing values for $Area_{UAi}/Area_{SAi}$ versus run-on as calculated by Equation 1 *or* $Area_{UAi}/Area_{Sai}$ versus infiltration rate as predicted by the HELP model to obtain an estimate of the mean run-on or infiltration rate for the mean value of $Area_{UAi}/Area_{SAi}$. The Python model interpolates between points in the lookup table assuming a linear relationship between $Area_{UAi}/Area_{SAi}$ and subsided-area run-on or infiltration rate. An example lookup table is shown in Figure 2.

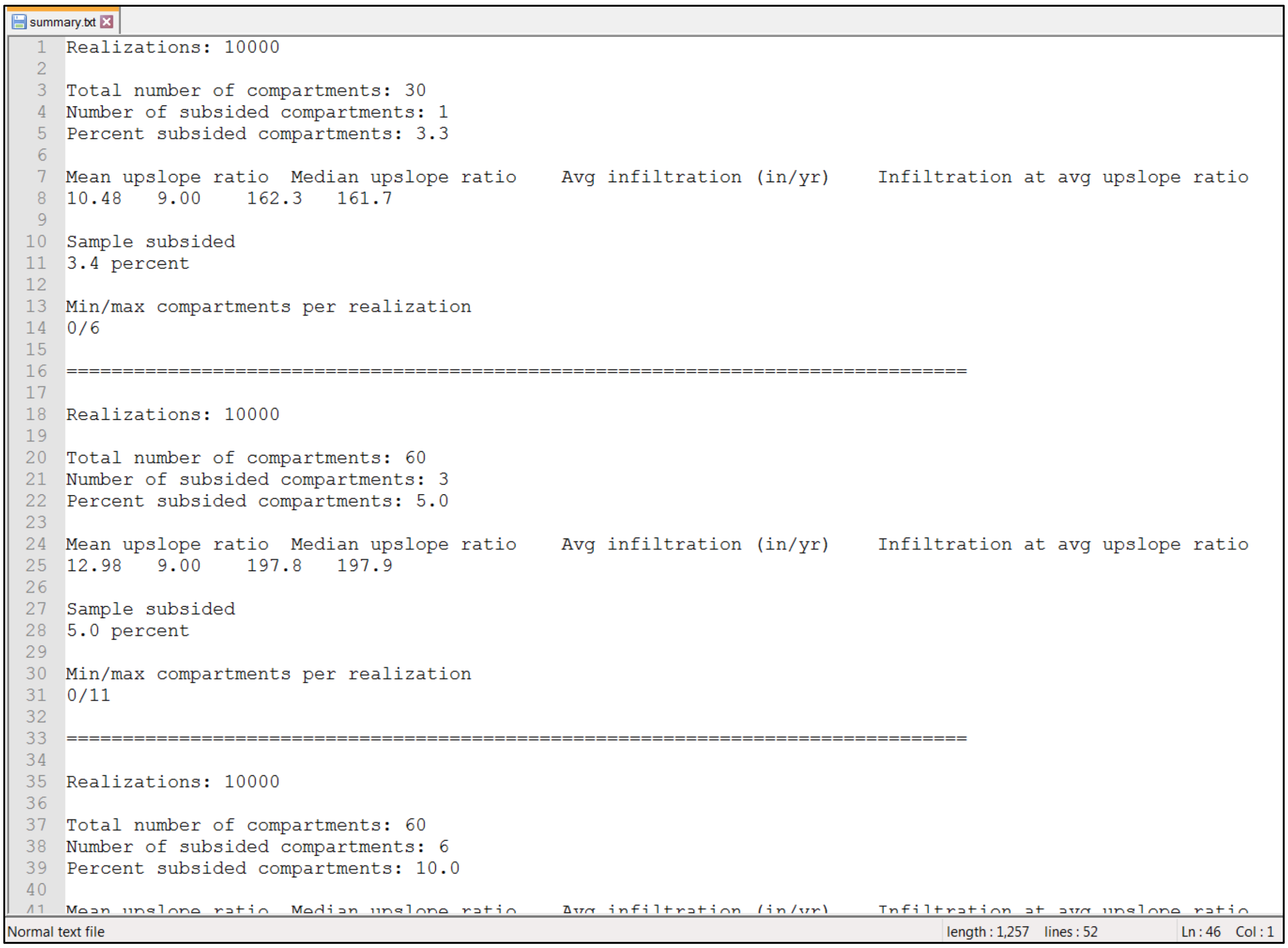
table.tab

```
0   15.9
0.1 17.1
0.5 21.5
1   27.7
2.5 48.6
5   84.0
10  154.9
20  299.0
25  367.7
30  438.6
60  864.2
```

Figure 2. Lookup Table for $Area_{UAi}/Area_{SAi}$ (Col. 1) vs. Infiltration Rate (inches/year) (Col. 2)

The lookup table is specific to the closure cap design of interest (i.e., percent slope, slope length, material and hydraulic properties of each layer, etc.) and is based on separate HELP model runs for both intact and subsided cases. As such, the lookup table must be updated by the user, and is dependent on percent slope and slope length. If the lookup table consists of $Area_{UAi}/Area_{SAi}$ versus run-on, model-calculated run-on rates from Equation (1), which are based on the predicted mean value of $Area_{UAi}/Area_{SAi}$, will then be separately input in the HELP model to obtain the desired subsided-area infiltration rates. On the other hand, if the lookup table comprises $Area_{UAi}/Area_{SAi}$ versus HELP-model-predicted infiltration rate, the probabilistic model will return subsided-area infiltration rates directly. In the latter approach, prerequisite HELP model runs are necessary for both the intact and subsided cases to generate the lookup table.

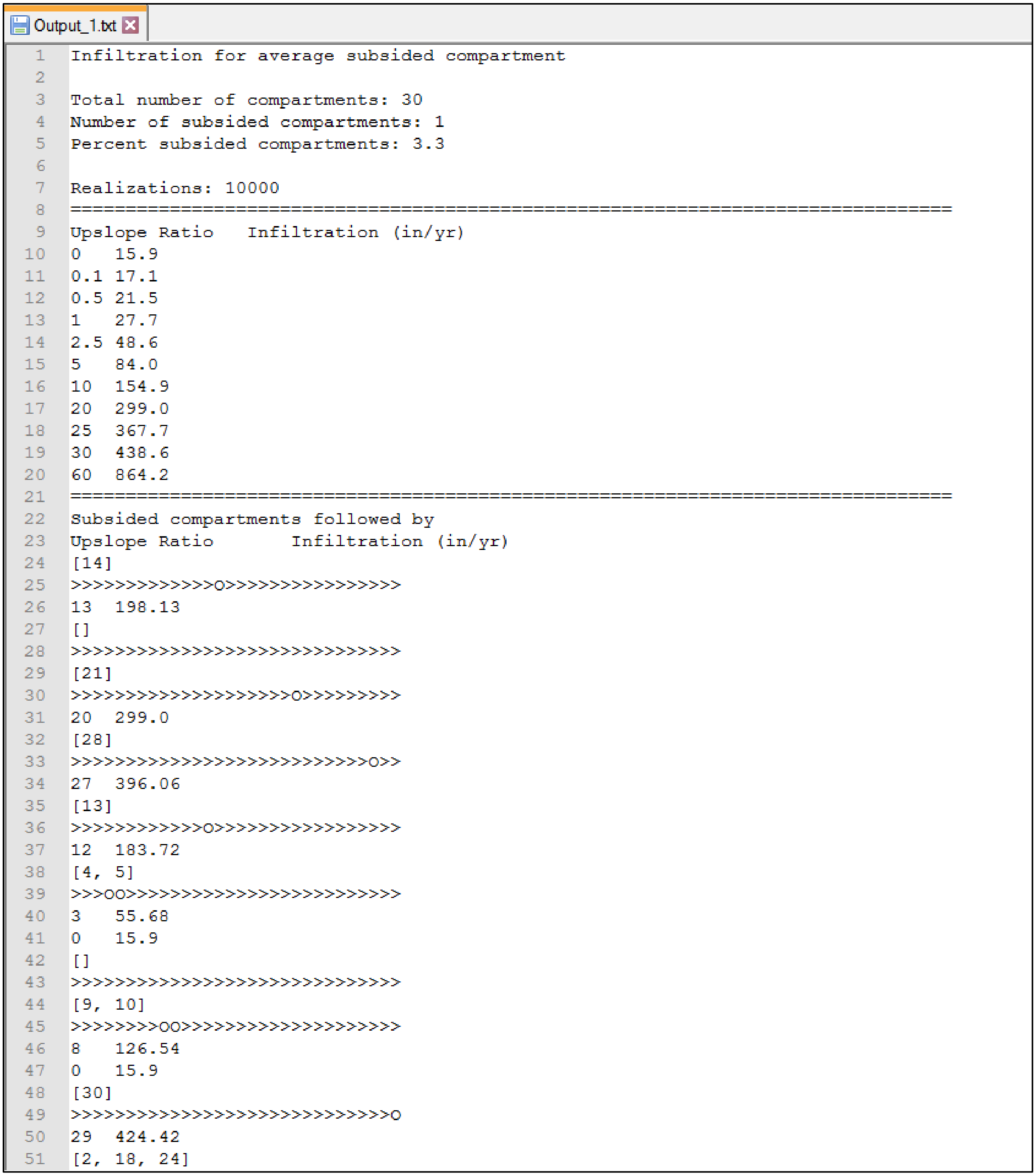
Screen captures of a partial summary file for all cases and a partial output file for an individual case are shown in Figure 3 and Figure 4, respectively.


summary.txt

```
Realizations: 10000

Total number of compartments: 30
Number of subsided compartments: 1
Percent subsided compartments: 3.3

Mean upslope ratio  Median upslope ratio    Avg infiltration (in/yr)    Infiltration at avg upslope ratio
10.48   9.00    162.3   161.7

Sample subsided
3.4 percent

Min/max compartments per realization
0/6

================================================================================

Realizations: 10000

Total number of compartments: 60
Number of subsided compartments: 3
Percent subsided compartments: 5.0

Mean upslope ratio  Median upslope ratio    Avg infiltration (in/yr)    Infiltration at avg upslope ratio
12.98   9.00    197.8   197.9

Sample subsided
5.0 percent

Min/max compartments per realization
0/11

================================================================================

Realizations: 10000

Total number of compartments: 60
Number of subsided compartments: 6
Percent subsided compartments: 10.0

Mean upslope ratio  Median upslope ratio    Avg infiltration (in/yr)    Infiltration at avg upslope ratio
```

Normal text file   length : 1,257   lines : 52   Ln : 46   Col : 1

Figure 3. Screen Capture of Partial Summary File for Multiple Case Simulation



```
Output_1.txt
Infiltration for average subsided compartment

Total number of compartments: 30
Number of subsided compartments: 1
Percent subsided compartments: 3.3

Realizations: 10000
================================================================================
Upslope Ratio   Infiltration (in/yr)
0    15.9
0.1  17.1
0.5  21.5
1    27.7
2.5  48.6
5    84.0
10   154.9
20   299.0
25   367.7
30   438.6
60   864.2
================================================================================
Subsided compartments followed by
Upslope Ratio       Infiltration (in/yr)
[14]
>>>>>>>>>>>>>O>>>>>>>>>>>>>>>>
13   198.13
[]
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
[21]
>>>>>>>>>>>>>>>>>>>>O>>>>>>>>>
20   299.0
[28]
>>>>>>>>>>>>>>>>>>>>>>>>>>>O>>
27   396.06
[13]
>>>>>>>>>>>>O>>>>>>>>>>>>>>>>>
12   183.72
[4, 5]
>>>OO>>>>>>>>>>>>>>>>>>>>>>>>>
3    55.68
0    15.9
[]
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
[9, 10]
>>>>>>>>OO>>>>>>>>>>>>>>>>>>>>
8    126.54
0    15.9
[30]
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>O
29   424.42
[2, 18, 24]
```

Figure 4. Screen Capture of Partial Output File for Individual Case in Multiple Case Simulation

Impact of the Number of Intact and Subsided Compartments

The probabilistic model as constructed uses uniformly sized dimensionless compartments to represent the intact and subsided areas along the length of the closure cap slope. As such, the predicted value of $Area_{UAi}/Area_{SAi}$ is independent of cap slope length and percent slope, and is strictly a function of the assumed total number of compartments (intact plus subsided) and the percent subsidence (number of randomly spaced subsided compartments/total number of compartments).

Table 2 and Figure 5 present results for both 5% and 10% subsidence. Note how the curves for $Area_{UAi}/Area_{SAi}$ begin to plateau once the total number of compartments exceeds several hundred, and eventually approach the conservative upper bound values shown in Table 1.

Table 2. Mean Upslope-Intact-Area to Subsided-Area Ratios for 5% and 10% Subsidence (10,000 realizations)

Total Number of Compartments	Upslope-Intact-Area to Subsided-Area Ratio for 5% Subsidence	Upslope-Intact-Area to Subsided-Area Ratio for 10% Subsidence
20	6.8	5.1
30	9.2	6.2
40	10.8	6.8
60	12.9	7.5
120	15.9	8.2
200	17.1	8.6
300	17.8	8.7
600	18.4	8.9
800	18.6	8.9
1,000	18.6	8.9
2,000	18.8	9.0
6,000	18.9	9.0
Limit	19.0	9.0

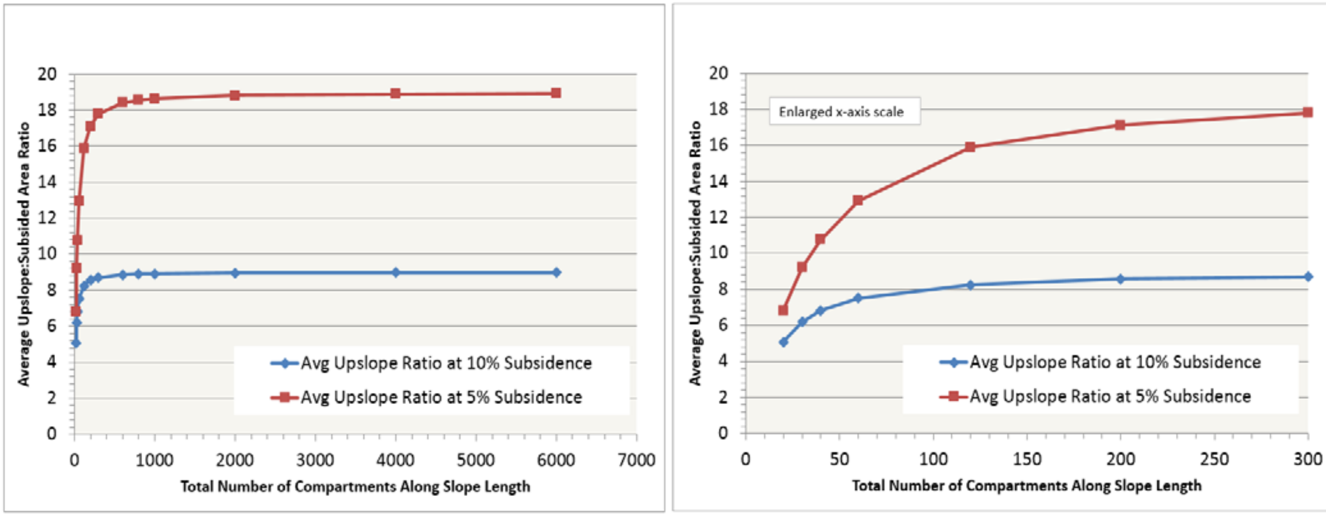


Figure 5. Mean Upslope-Intact-Area to Subsided-Area Ratio for 5% and 10% Subsidence (10,000 realizations)

Practical Considerations

Invoking practical considerations for subsided compartment "hole" size as well as the number of subsided compartments or "holes" substantially shrinks the parameter space to be explored by the vadose-zone model. Figure 6 displays a plot of hole size versus percent subsidence (or percent non-crushable) as a function of the number of holes (NH) for a representative 600-foot slope length. Assuming a realistic subsided-area hole size of ≥ 10 feet, a non-crushable content ≤ 10 percent based on historical operation constraints for slit trenches, and a minimum of 1 hole to be considered a subsided case, the relevant parameter space shrinks to the red triangular zone shown in Figure 6 and Figure 7. This relevant parameter space covers the following ranges: one to six holes, hole sizes of 10 to 60 feet, and 1.6% to 10% subsidence (shaded region in Table 1).

The blue circle within the relevant parameter space represents a proposed central case to be explored in the E-Area LLWF PA deterministic and/or uncertainty analyses: two 20-foot-sized holes corresponding to 6.7% subsidence over a 600-foot slope length. The maximum value for $Area_{UAi}/Area_{SAi}$ is 14 per Table 1, and occurs only if one of the holes is conservatively located at the toe (bottom edge) of the cap. This bounding scenario is not likely to occur. Conversely, the mean or best-estimate value for $Area_{UAi}/Area_{SAi}$ based on 100,000 Monte Carlo simulations of randomly spaced subsided compartments is 7.9.

Assuming future operations resemble past disposal practices, the expected subsidence level can be estimated from the non-crushable waste container content in slit trenches filled to at least 90% by volume: SLIT1 (0.0%), SLIT2 (3.1%), SLIT3 (7.7%), SLIT4 (5.7%), SLIT5 (0.9%), SLIT6 (2.9%), and SLIT8 (0.0%). The arithmetic mean of the non-zero non-crushable data for slit trenches is approximately 4%, which is slightly biased toward the lower end of the historical operational window of 0% to 10%. The conservative upper bound value for $Area_{UAi}/Area_{SAi}$ at 4% subsidence is 24 (Table 1), while the mean or best-estimate value for $Area_{UAi}/Area_{SAi}$ assuming one randomly located 24-foot hole is 8.7 (100,000 Monte Carlo realizations using Python probabilistic model). Consideration of this 4% subsidence case is also proposed for the E-Area LLWF PA deterministic and/or uncertainty analyses.

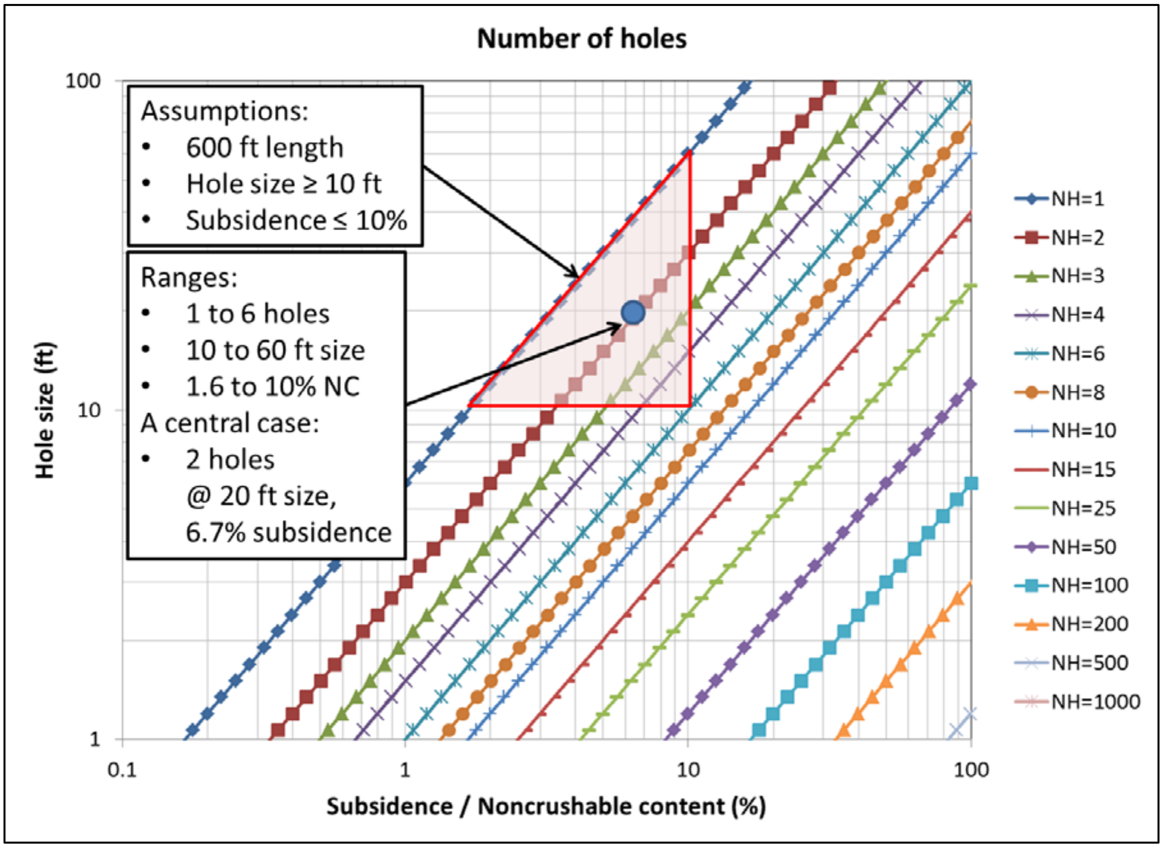


Figure 6. Subsided Compartment Parameter Space for 600-foot Slope Length

<u>Recommendations</u>

For the next revision of the E-Area PA:

- Based on historical data for non-crushable content in E-Area slit trenches, a 4% subsidence (mean of non-zero data) case should be considered using a conservative upper bound upslope-intact-area to subsided-area ratio ($Area_{UAi}/Area_{SAi}$) of 24 (Table 1) and/or best-estimate ratio of 8.7 (one 24-foot hole) depending on the desired level(s) of conservatism.

- Because future practices may differ from past disposals, a 6.7% subsidence case with two 20-foot subsided compartments is also recommended. This case lies near the center of the current and expected future E-Area trench operational window. The conservative upper bound of $Area_{UAi}/Area_{SAi}$ for this case is equal to 14, and the best estimate is 7.9.

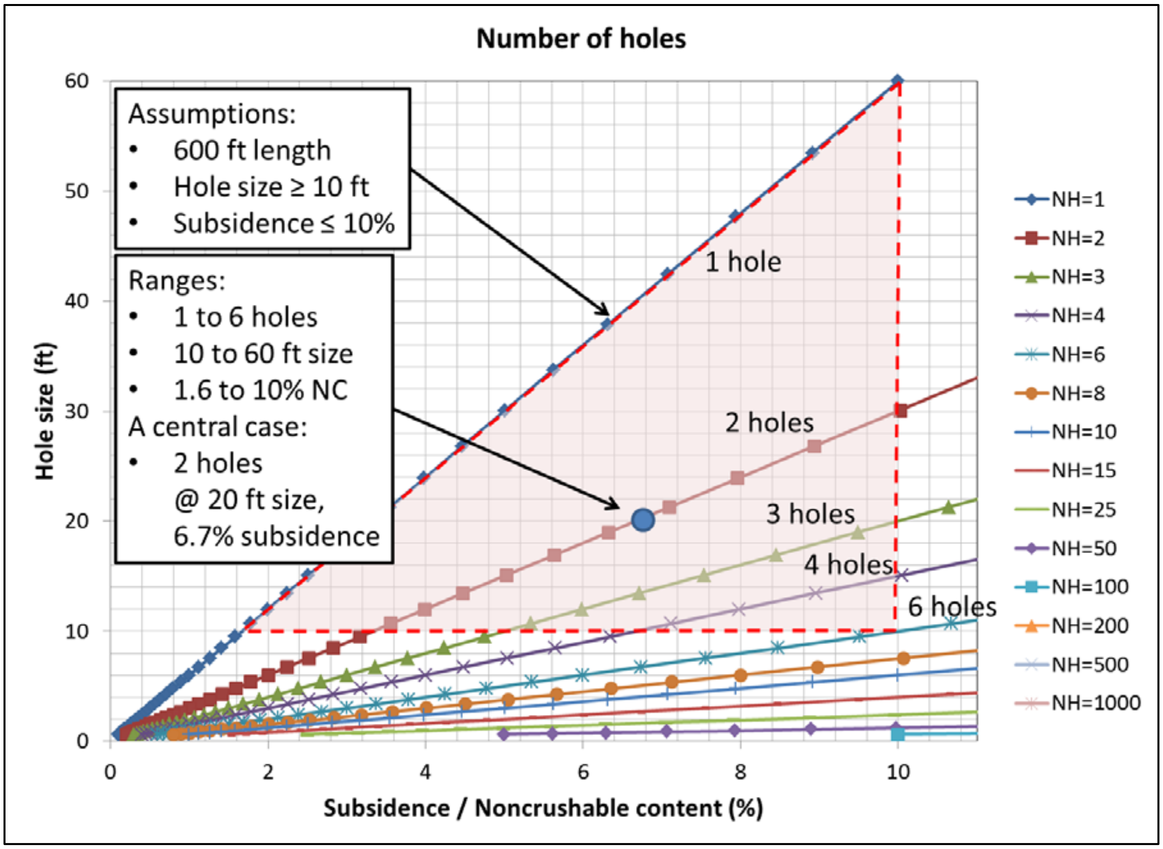


Figure 7. Relevant Subsided Compartment Parameter Space for Vadose Zone Modeling

References

Dyer, J. A. (2017a) Conceptual Modeling Framework for E-Area PA HELP Infiltration Model Simulations. SRNL-STI-2017-00678. Savannah River National Laboratory, Aiken, SC.

Dyer, J. A. (2017b) E-Area Low-Level Waste Facility Vadose Zone Model: Confirmation of Water Mass Balance for Subsidence Scenarios. SRNL-STI-2017-00728. Savannah River National Laboratory, Aiken, SC.

Flach, G. P. (2017) Overall Conceptual Approach to Uncertainty Quantification and Sensitivity Analysis for the E-Area PA Revision. SRNL-STI-2017-00062. Savannah River National Laboratory, Aiken, SC.

This Page Intentionally Blank

Appendix A

Python Source Code for Probabilistic Model

```
#!/bin/env python

import sys
import random
import numpy
import matplotlib.pyplot as plt


print "Running:", sys.argv[0]


### Read filenames
compartmentsTotal    = sys.argv[1]   # total number of compartments
compartmentsSubsided = sys.argv[2]   # total number of subsided compartments
realizations         = sys.argv[3]   # number of realizations
tableFile            = sys.argv[4]   # name of table file (average area ratio vs. subsided infiltration rate)
outputFile           = sys.argv[5]   # name of output file
summaryFile          = sys.argv[6]   # name of summary file
debugArg             = sys.argv[7]   # debug flag (0 is off and 1 is on)
appendFlag           = sys.argv[8]   # append flag for summary file

if debugArg == "True":
    debugFlag = 1
else:
    debugFlag = 0

percentSubsided = float(compartmentsSubsided)/float(compartmentsTotal)*100

print "   compartmentsTotal:", compartmentsTotal
print "compartmentsSubsided:", compartmentsSubsided
print "  percentSubsided:", percentSubsided
print "           realizations:", realizations
print "     tableFile:", tableFile
print "        outputFile:", outputFile
print "            summaryFile:", summaryFile

output = open(outputFile, 'w')
output.write("Infiltration for average subsided compartment\n\n")
output.write("Total number of compartments: %s\n" % (compartmentsTotal))
output.write("Number of subsided compartments: %s\n" % (compartmentsSubsided))
output.write("Percent subsided compartments: %.1f\n\n" % (percentSubsided))
output.write("Realizations: %s\n" % (realizations))
```

```
### Read table: (upgradient length, subsided infiltration)
infiltrationTable = {}
n = 0

with open(tableFile) as file:
    for line in file:
       if line.strip():
           n = n + 1
           #print n
           (upgradientLength,subsidedInfiltration) = line.split("   ")
           infiltrationTable[upgradientLength] = subsidedInfiltration.strip()
file.close()
print "       table records:", n

if debugFlag: print sorted(infiltrationTable)
if debugFlag: print sorted(infiltrationTable, key=float)
output.write("================================================================================\n")
output.write("Upslope Ratio\tInfiltration (in/yr)\n")
for key in sorted(infiltrationTable, key=float):
    output.write("%s\t%s\n" % (key,infiltrationTable[key]))


xp = sorted(infiltrationTable, key=float)
if debugFlag: print "xp:", xp

yp = []
for key in xp:
    yp.append(infiltrationTable[key])
if debugFlag: print "yp:", yp

plt.plot(xp,yp,'ro-')
plt.xlabel('Upslope ratio')
plt.ylabel('Infiltration (in/yr)')
plotName = tableFile + '.png'
plt.savefig(plotName)
#if debugFlag: plt.show()


### Subsided trench compartments
output.write("================================================================================\n")
output.write("Subsided compartments followed by\n")
output.write("Upslope Ratio\t\tInfiltration (in/yr)\n")

compartments = range(1,int(compartmentsTotal)+1) #generates sequence of numbers as a list (e.g., [1,2,3,...60]; value of i+1 not included in range
if debugFlag: print "compartments:", compartments
```

```
stringIntact = [] # creates empty list named stringIntact
for i in compartments: #i could be any name, e.g., i, jim, cow, key
    stringIntact.append(">")
print stringIntact

realization = 0
avgLength = float(0) # floating point real number
avgInfiltration = float(0)
avgCount = 0

countSubsided = 0
countTotal = 0

maxCompartments = 0
minCompartments = compartmentsTotal
upslopeArea = []

while realization < int(realizations):
    realization = realization + 1
    if debugFlag: print realization

    # prior logic:
    #subsided = random.sample(compartments, int(compartmentsSubsided))

    subsided = []
    localCount = 0
    for i in compartments:
        draw = random.uniform(0,100)
        countTotal = countTotal + 1
        if draw < percentSubsided:
            countSubsided = countSubsided + 1
            subsided.append(i)
            localCount = localCount + 1
    maxCompartments = max(maxCompartments, localCount)
    minCompartments = min(minCompartments, localCount)

    if debugFlag: print "subsided:", subsided

#    sortedSubsided = sorted(subsided, key=float)
#    if debugFlag: print "sortedSubsided:", sortedSubsided

    stringSubsided = list(stringIntact)
    for key in subsided:
        stringSubsided[int(key)-1] = 'O'
    if debugFlag: print "stringSubsided:", "".join(stringSubsided)
```

```
    lengthSubsided = []
    infiltrationSubsided = []

    for i in range(len(subsided)):  # loops over numbers [0,1,2] for 3 subsided cells
        if i == 0:
            length = subsided[i] - 1
        else:
            length = subsided[i] - subsided[i-1] - 1

        infiltration = numpy.interp(length,xp,yp)

        if debugFlag: print "i, subsided[i], length, infiltration:", i, subsided[i], length, infiltration

        lengthSubsided.append(length)
        infiltrationSubsided.append(infiltration)
        upslopeArea.append(length)
        sortedUpslopeArea = sorted(upslopeArea, key=float)

    if debugFlag: print "lengthSubsided:", lengthSubsided
    if debugFlag: print "infiltrationSubsided:", infiltrationSubsided
    if debugFlag: print "sorted upslopeArea:", sortedUpslopeArea

    print >> output, subsided
    print >> output, "".join(stringSubsided)

    for i in range(len(infiltrationSubsided)):
        avgCount = avgCount + 1
        avgLength = avgLength + lengthSubsided[i]
        avgInfiltration = avgInfiltration + infiltrationSubsided[i]
        output.write("%s\t%s\n" % (lengthSubsided[i], infiltrationSubsided[i]))

avgLength = avgLength/float(avgCount)
avgInfiltration = avgInfiltration/float(avgCount)
infiltrationAtAvgLength = numpy.interp(avgLength,xp,yp)
upslopeAreaMean = numpy.mean(sortedUpslopeArea)
upslopeAreaMedian = numpy.median(sortedUpslopeArea)

sampleSubsided = float(countSubsided)/float(countTotal)*100


output.write("================================================================================\n")
output.write("Percent subsided\tMean upslope ratio\tMedian upslope ratio\tAvg infiltration (in/yr)\tInfiltration at avg upslope ratio\n")
output.write("%.1f\t\t\t%.2f\t\t\t%.2f\t\t\t%.1f\t\t\t%.1f\n\n" % (percentSubsided, upslopeAreaMean, upslopeAreaMedian, avgInfiltration,
infiltrationAtAvgLength))
output.write("Sample subsided\n")
output.write("%.1f percent\n\n" % (sampleSubsided))
output.write("Min/max compartments per realization\n")
```

```
output.write("%d/%d\n\n" % (minCompartments, maxCompartments))

print ("percentSubsided, avgLength, mean upslope area ratio, median upslope area ratio: \n%.1f %.2f %.2f %.2f" % (percentSubsided, avgLength,
upslopeAreaMean, upslopeAreaMedian))
print ("avgInfiltration, infiltrationAtAvgLength: \n%.1f %.1f" % (avgInfiltration, infiltrationAtAvgLength))
print ("PercentSubsided Sample: %.1f" % (sampleSubsided))

output = open(summaryFile, appendFlag)
output.write("Realizations: %s\n\n" % (realizations))
output.write("Total number of compartments: %s\n" % (compartmentsTotal))
output.write("Number of subsided compartments: %s\n" % (compartmentsSubsided))
output.write("Percent subsided compartments: %.1f\n\n" % (percentSubsided))
output.write("Mean upslope ratio\tMedian upslope ratio\tAvg infiltration (in/yr)\tInfiltration at avg upslope ratio\n")
output.write("%.2f\t%.2f\t%.1f\t%.1f\n\n" % (upslopeAreaMean, upslopeAreaMedian, avgInfiltration, infiltrationAtAvgLength))
output.write("Sample subsided\n")
output.write("%.1f percent\n\n" % (sampleSubsided))
output.write("Min/max compartments per realization\n")
output.write("%d/%d\n\n" % (minCompartments, maxCompartments))
output.write("=================================================================================\n\n")
```

This Page Intentionally Blank

This Page Intentionally Blank

Distribution List

timothy.brown@srnl.doe.gov
alex.cozzi@srnl.doe.gov
david.crowley@srnl.doe.gov
david.dooley@srnl.doe.gov
a.fellinger@srnl.doe.gov
samuel.fink@srnl.doe.gov
jeff.griffin@srnl.doe.gov
nancy.halverson@srnl.doe.gov
connie.herman@srnl.doe.gov
john.mayer@srnl.doe.gov
daniel.mccabe@srnl.doe.gov
frank.pennebaker@srnl.doe.gov
luke.reid@srnl.doe.gov
boyd.wiedenman@srnl.doe.gov
bill.wilmarth@srnl.doe.gov
sebastian.aleman@srnl.doe.gov
tom.butcher@srnl.doe.gov
Thomas.Danielson@srnl.doe.gov
kenneth.dixon@srnl.doe.gov
James.Dyer@srnl.doe.gov
gregory.flach@srnl.doe.gov
lee.fox@srs.gov
luther.hamm@srnl.doe.gov
thong.hang@srnl.doe.gov
ginger.humphries@srs.gov
ralph.nichols@srnl.doe.gov
Roger.Seitz@srnl.doe.gov
Ira.Stewart@srs.gov
kevin.tempel@srs.gov
Tad.Whiteside@srnl.doe.gov
Jennifer.Wohlwend@srnl.doe.gov
T. N. Foster, EM File, 773-42A – Rm. 243
(1 file copy and 1 electronic copy)
Records Administration (EDWS)