Key Words:
Saltstone
Performance Assessment

Retention:
Permanent

Saltstone Disposition Facility Stochastic Transport and Fate Model Description

Glenn A. Taylor

MARCH 2009

Savannah River National Laboratory
Savannah River Nuclear Solutions
Aiken, SC 29808

Prepared for the U.S. Department of Energy Under
Contract Number DE-AC09-08SR22470

SRNL™

DISCLAIMER

Printed in the United States of America

Prepared for
U.S. Department of Energy

Key Words:
Saltstone
Performance Assessment

Retention:
Permanent

Saltstone Disposition Facility Stochastic Transport and Fate Model Description

Glenn A. Taylor

MARCH 2009

Savannah River National Laboratory
Savannah River Nuclear Solutions
Aiken, SC 29808

Prepared for the U.S. Department of Energy Under Contract Number DE-AC09-08SR22470

SRNL™

REVIEWS AND APPROVALS

Author

G. A. Taylor, Radiological Performance Assessment Date: 3/16/09

Technical Review

M. H. Layton, S3116 Documentation Date: 3/16/09

Approval

H. H. Burns, Project Lead, Radiological Performance Assessment Date: 3/17/09

D. A. Crowley, Manager, Radiological Performance Assessment Date: 3/19/09

S. L. Marra, Manager, E&CPT Research Programs Date: 3/20/09

1.0 Stochastic SDF Transport and Fate Model

The stochastic Saltstone Disposal Facility (SDF) model is described in this report. This model incorporates lessons-learned from the FTF modeling effort (FTF PA, 2008). Figure 1 shows the top level of the transport and fate model. The model is implemented in the GoldSim modeling environment. The various GoldSim containers will be discussed in the following sections. The discussion will focus on Vault2, but is applicable to all vault types. Differences existing in the modeling of the different vaults types will be described as necessary. The GoldSim element names will be referred to in italics.

A brief overview of the major model elements is:

- *TheVaults* – contains the elements necessary for contaminant transport
- *Materials* – contains the material definitions and properties used in *TheVaults*
- *Inventory* – contains the source term information
- *Transport* – contains the flows used in the *TheVaults* for contaminant transport
- *DoseAssessment* – contains all dose related data and calculations

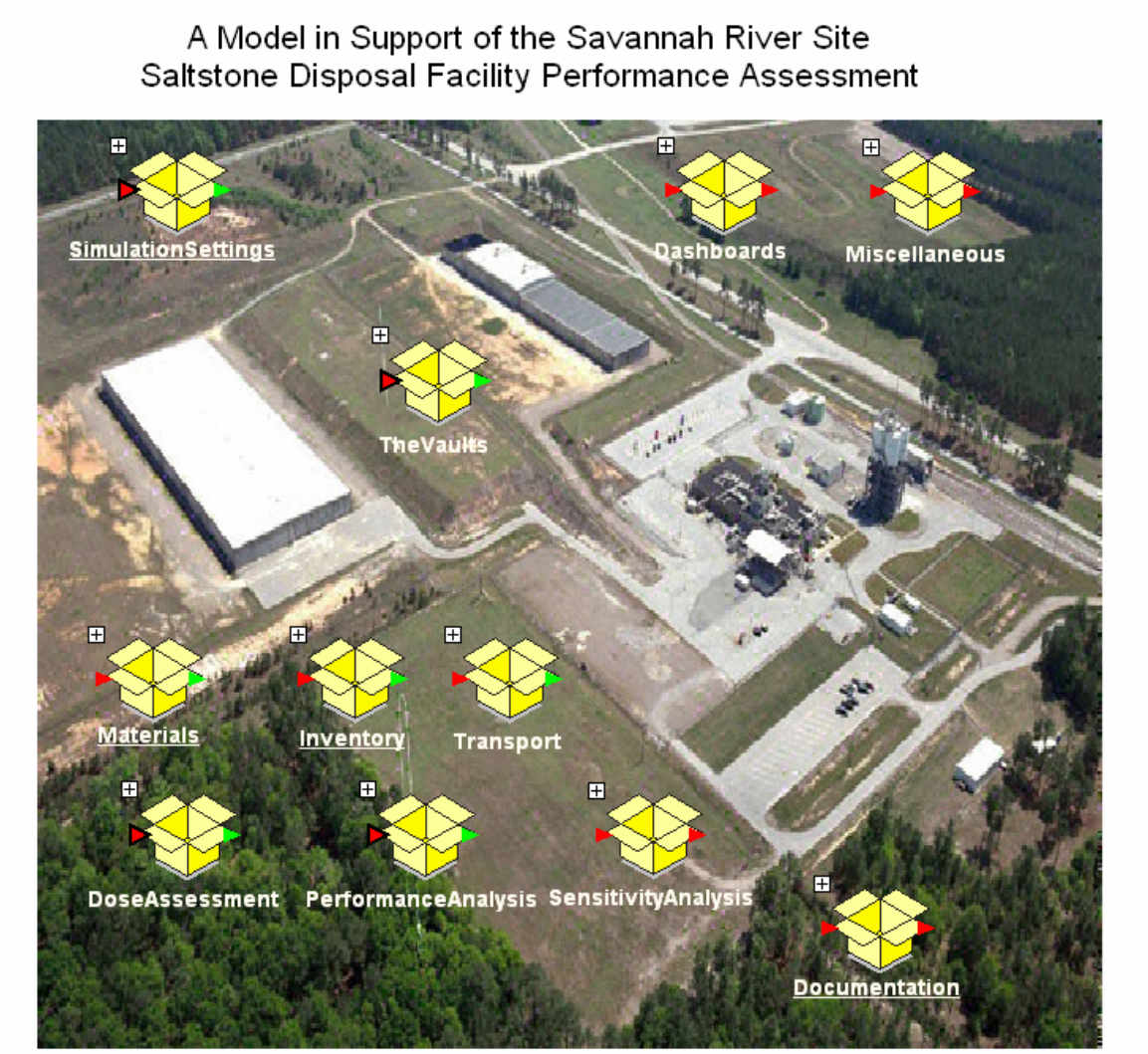

Figure 1 Top Level Transport and Fate Model

1.1 Modeling Philosophy

This section contains a brief description of why the model was designed as it is.

The primary driver in the design of this model is that it is a stochastic model. Its statistics are generated by using a Monte Carlo simulation which involve the running of the model many times. As such, the model must be designed such that it gives reasonable results in as short a run time as possible. Some of the short run-time requirement is met by the modeling environment being 1-dimensional and performing only a transport simulation, i.e., it does not perform a flow field calculation.

It was determined that the analysis to support the SDF PA would not use solubility limits. This provided an additional avenue through which run-times could be reduced. Solubility confers a non-linearity whereas a distribution coefficient (K_d) leaves one with a

linear formulation. This linear formulation allows for the modeling of only one Vault2. Results for all the other Vault2's can then be scaled from the results of the one calculation, thereby saving considerable computational time. The implications of these underlying paradigms will be discussed in detail in this report.

1.2 Container *TheVaults*

TheVaults, as shown in Figure 2, is where the transport of contaminants is calculated.

Figure 2 TheVaults Container

The containers *Vault_1, Vault_2, and Vault_4* contain the transport calculations for each of the vaults. *VaultData* contains the geometric data for all 3 vaults. *PlumeCalc_Sectors* contains the information necessary to calculation the plume function for each of the

disposal cells relative to the sectors they affect. This will be described in detail in Section 1.6

1.2.1 Container *Vault_2*

This section contains the main discussion on the transport section of the model. Any differences between the vaults will be noted, but in most cases those differences are minor. Detailed descriptions of this container's sub-elements follow.

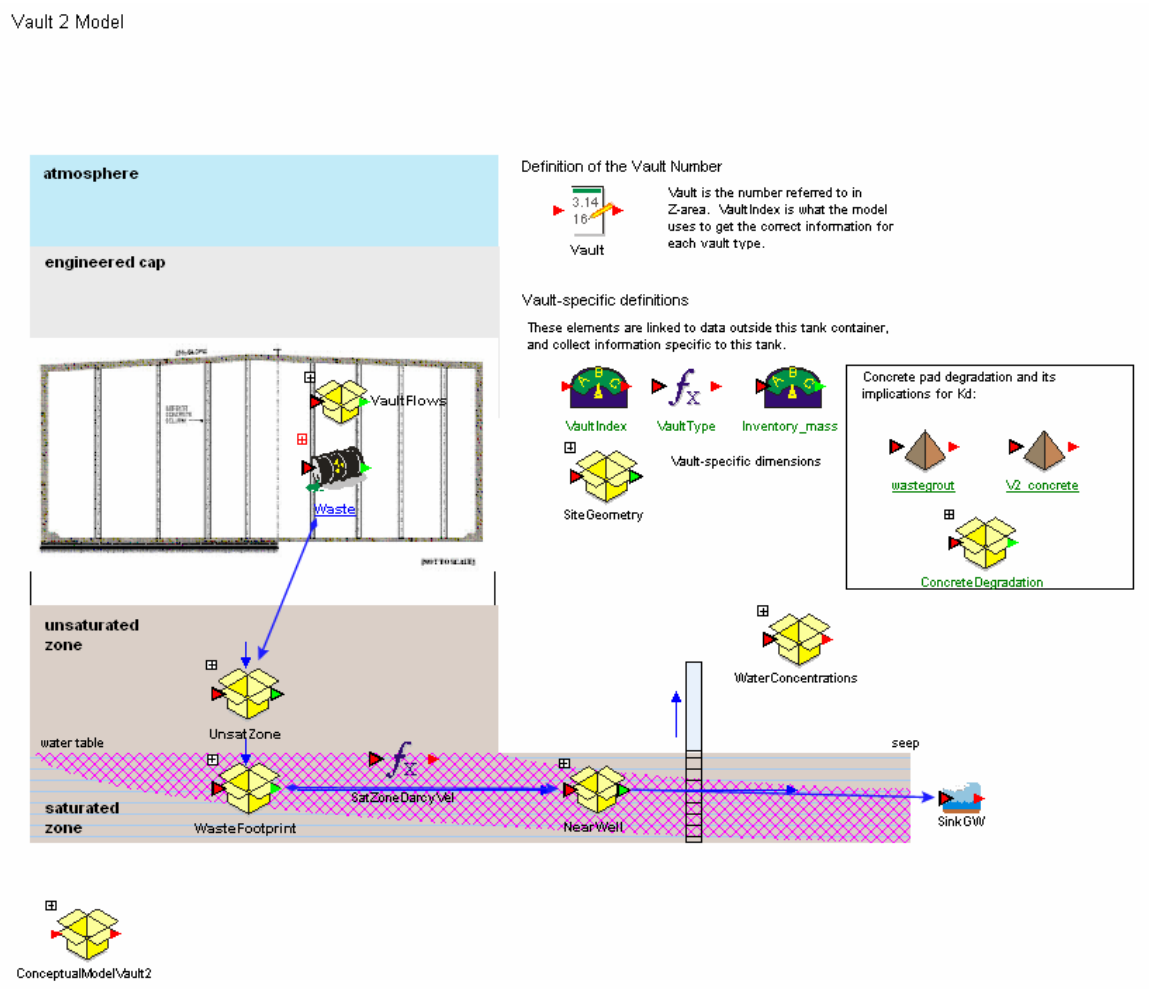

Figure 3 Vault2 Transport Module

1.2.1.1 Concrete Degradation

The concrete degradation model as implemented in the Goldsim model is one which affects only the chemistry of the concrete. The degradation affect on flow is implicitly

applied through the Porflow flow rates. The degradation affects the K_d's and is based on the number of pore water flushes [Denham, 2008]. A local definition of the solids *wastegrout* and *V2_concrete* is in each vault container. By doing this each vault can have it chemical properties change independently, but as always, all the Vault2's change at the same time. Porflow analyses showed that the floors of the three vaults did not experience the change from reducing to oxidizing in 20,000 years, therefore, that transition is not included in this model.

1.2.1.1.1 Container *ConcreteDegradation*

The container *ConcreteDegradation* (see Figure 4) holds the calculations of pore volume flushes which affect the K_d's of the cementitious materials. *PreviousValues* elements had to be used because the code reported a recursive loop. The *Selector* elements determine which set of K_d's are used based on *NumberofFlushes[Grout,Wall]*.

There are two data elements which are benchmarking fractions (BMF) applied to ^{99}Tc's K_d. ^{99}Tc was treated differently in Porflow from all the other radionuclides (see [Flach, 2009]). These BMF's are to account for those formulations and are discussed in detail in [Taylor, 2009].

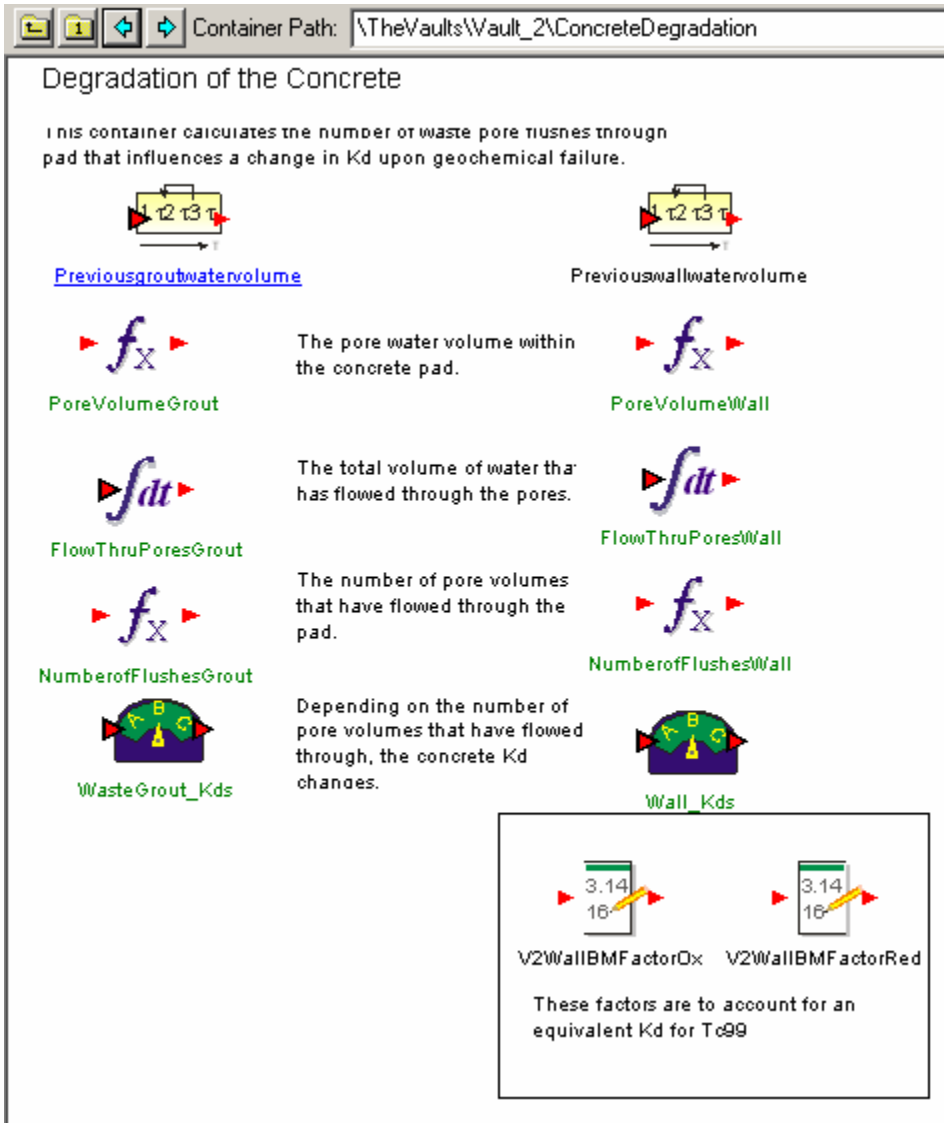

Figure 4 Container *ConcreteDegradation*

Figure 5 shows the elements which are evaluated by the selector elements to determine which set of K_d's are to be used by the grout and the wall. This was thought to be a potentially important phenomenon in controlling the waste release so the number of pore flushes for the reducing-oxdizing transtion was described by a *Stochastic* element. [Denham, 2008] did not contain an uncertainty analysis. The author of the report recommended using ±50%, and that is was is implemented by the model. Two transitions are considered. "1st" is the transition from middle reducing to middle oxidizing and "2nd" is from middle oxidizing to old oxidizing. The vault concrete has different K_d's than the grout, hence the two separate pathways.

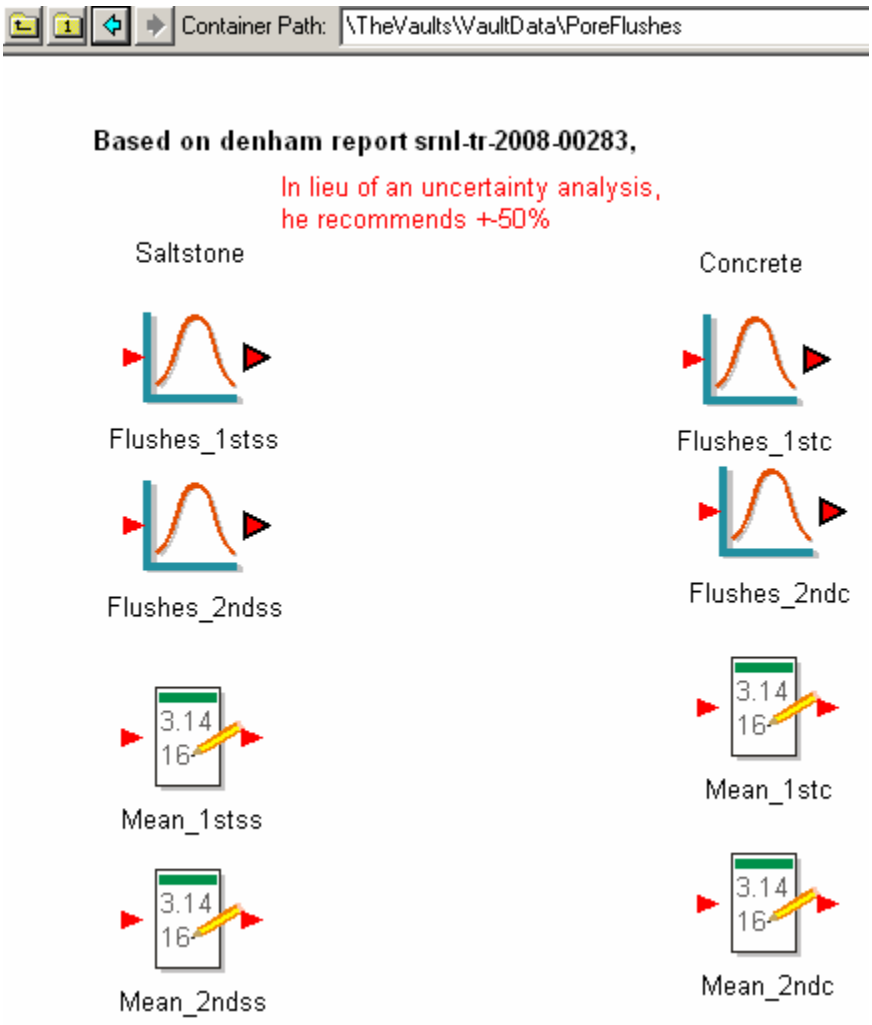

Figure 5 Container *PoreFlushes*

1.2.1.2 Container *VaultFlows*

The contents of *VaultFlows* are shown in Figure 8. A brief description of the elements follows. This element determines the flow through the waste zone.

1.2.1.2.1 Element *Configuration*

This element is the probability of occurrence of different Case as shown in Figure 6 and Figure 7.

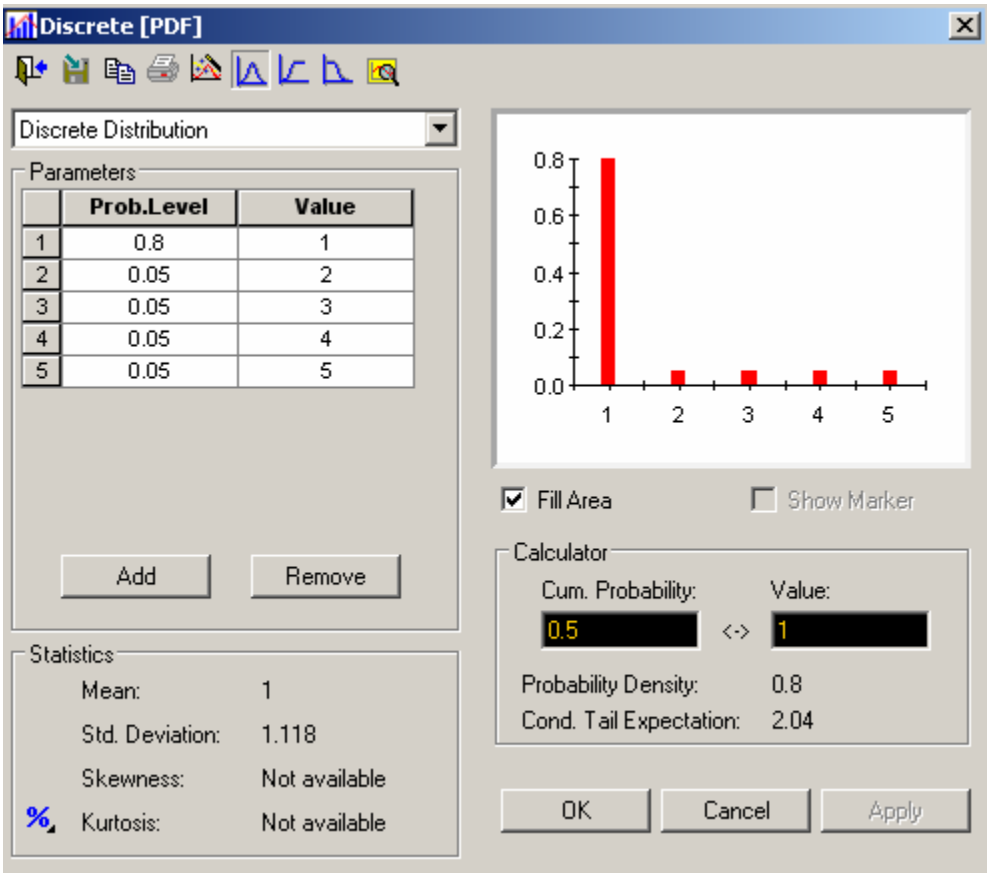

Figure 6 Vault2 and Vault4 Configuration Stochastic Element

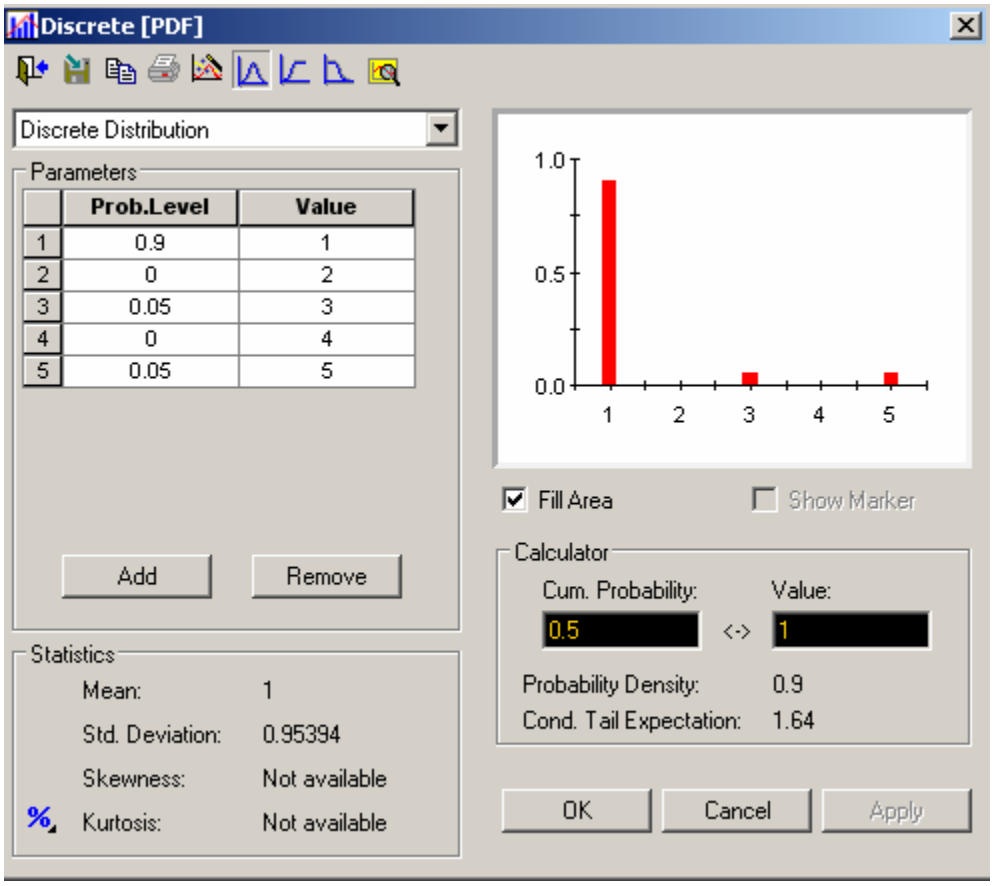

Figure 7 Vault1 Configuration Stochastic Element

1.2.1.2.2 Element *TimePeriodSelector*

This element determines which of the PorFlow time periods the calculation is in.

1.2.1.2.3 Elements *PF_Flow[grout,Dirt,UZ,Wall]*

This element picks the infiltration from the tables in *Transport* by choosing the appropriate time period and Case. The [*grout, Dirt, UZ*] elements refer to vectors such that a vertical gradient of the flow is supplied to the model. The [*Wall*] is a single value as an examination of the PorFlow results showed this to be a reasonable assumption. A detailed discussion of the abstraction of the flow data is in Section 1.3.

1.2.1.2.4 Element *UnsatDarcyFlux[Grout, Dirt, UZ, Wall]*

This element determines whether to use a natural infiltration rate or a PorFlow generated flowrate.

Figure 8 Element *VaultFlows*

1.2.2 Element *Waste*

Waste (Figure 9) contains the transport cells at the waste form elevations in *VaultCells*. *BypassFraction* is a stochastic on how much flow bypasses the waste form

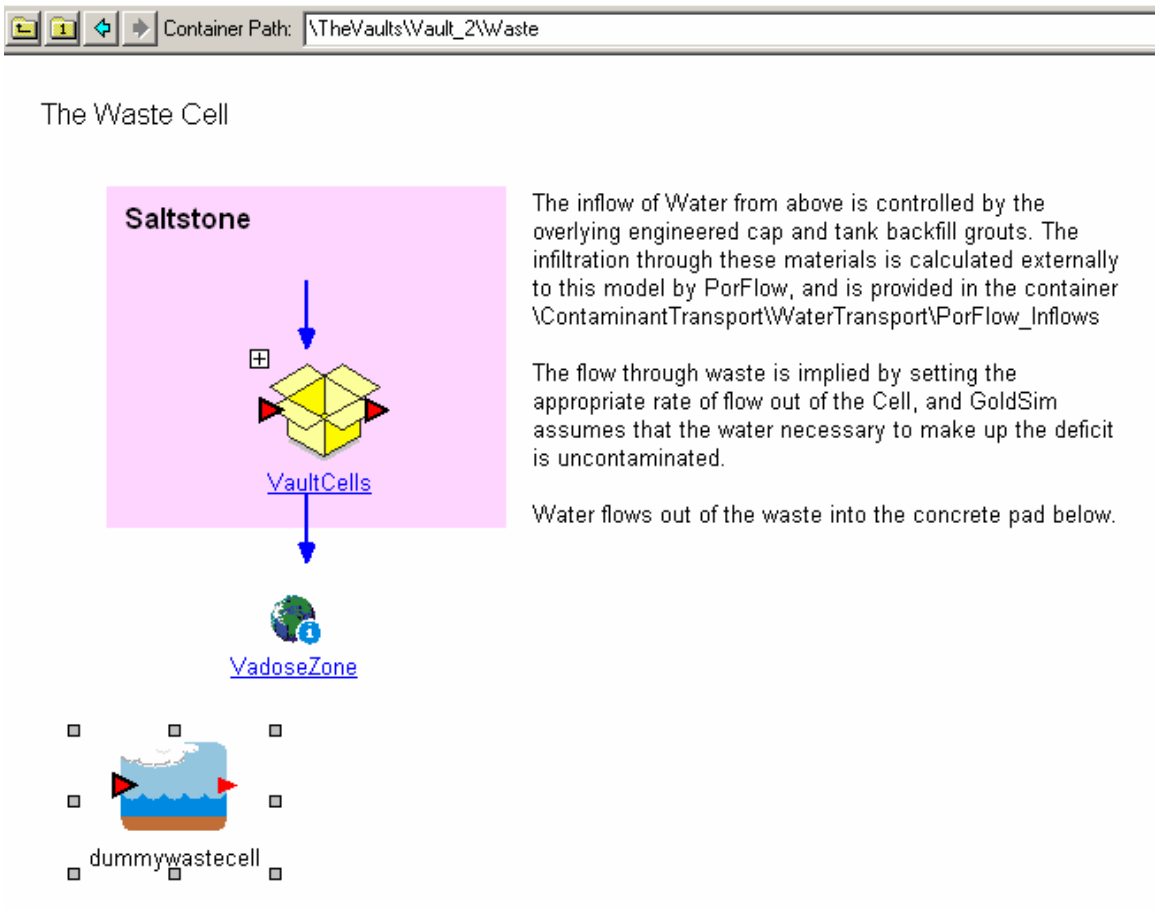

Figure 9 Element *Waste*

1.2.3 Container *VaultCells*

VaultCells (Figure 10) is where the transport of contaminants from the waste is performed and is the "top" of the transport model. The top boundary of the model is the top of the waste (saltstone). Therefore, the flow boundary conditions are applied at this level.

One will observe that there are three parallel flow paths in this model, *Grout*NN , *VW*NN *Dirt*NN where "NN" corresponds to an computational cell (in this case, a GoldSim mixing cell element). These three paths are necessary for several reasons. First, diffusion is an important element of waste transport and must be considered. Second, because diffusion is considered, the flow rates in the grout, wall, and dirt are quite different which results in vastly different transport rates of the contaminants. As a result of this conceptual model formulation, the model will be referred to as a pseudo-2D model.

1.2.3.1 Grout Cells

The Grout cells are those cells which contain saltstone. The cells transport contaminants by advection and diffusion. The advective transport is done in the classic 1-D methodology and accounts for transport in the z-direction. The diffusive transport is done in the r-direction (for a Vault2 design) or the x-direction (for Vault1 and Vault4) and this is what gives the model its pseudo-2D behavior.

The Grout cells' advection is accomplished by applying a flow boundary condition abstracted from PorFlow. A discussion of the flow abstraction can be seen in Section 1.3. Although the flow is fairly uniform in the grout monolith, a flow gradient is maintained in the z-direction.

The Grout diffusion is accomplished by the use of an analytical model of a shrinking core. The solution specifies the amount of mass which must be transferred each time step from the grout to the vault wall and dirt. Details of the model may be seen in Section 1.5.

This diffusion method was chosen over implementation of the GoldSim diffusion paradigm for several reasons. First, there are some issues with the implementation of the GoldSim diffusion model. The model requires a tortuosity, and that is a term which is quite difficult to define in modeling space. This is an issue with all diffusion models, not just the GoldSim implementation. Second, is the issue of discretization. A nodalization study was undertaken and it showed that approximately 400 cells would be necessary to reasonably calculate the diffusive term. This number of coupled cells would make for long run times, on the order of minutes rather than seconds, for one realization. Because this model is designed to be the uncertainty and sensitivity model, it must be able to run many realizations. In order to run a stochastic analysis, of say 10,000 realizations, one would be looking at perhaps weeks rather than overnight. Third, the important stochastic parameters in the diffusion equation appear in both the analytic solution and the GoldSim implementation. Therefore, one can assess those parameters equally well in either formulation.

1.2.3.2 Dirt Cells

The dirt cells represent those cells contiguous to the waste zone to which diffusion may occur. Unlike the grout cells, the flow in the dirt cells can vary considerably. This is because of the cap drain which channels the runoff into this region. The flow varies greatly in the z-direction and not nearly as much in the r-direction. Each dirt cell has its own flow rate. Note that in GoldSim, flow is important only as a mechanism by which the contaminant mass may be transported. No mass balance is performed on the water as its use it to essentially provide the timing of the transport (if one is not concerned about the concentration, which in this instance is not a concern). The dirt cells' volume is given as a unit volume. As just mentioned, at this point concentration is not a concern, only the transport. The concentration will be accounted for once the dirt cells dump into the unsaturated zone cells beneath the vault. The output of the bottom cell of the dirt

column, *VaultDirt20*, connects directly with the inlet cell of the *UnsatZone*, bypassing the vault floor.

Figure 10 Container *VaultCells*

1.2.3.3 Initial inventories

There are two initial inventories, one for the saltstone and one for the vault walls. The initial Vault2 wall inventory is zero as radionuclides are not expected to diffuse through the cells' liners. Vaults 1 and 4 have an initial wall inventory based on the benchmarking. The fractions of inventory in the wall and saltstone add up to 1.

The total initial inventory for each radionuclide in each disposal cell is 1 Ci. With the linear K_d model, the actual, or expected inventory, depending on disposal cell, is determined as shown in Section 1.7.1.

1.2.3.4 Container *VaultFloor*

Vaultfloor is in the model for two reasons; it provides for a different material type than the grout, and, it can have a different flow than the grout. It is the area directly under the saltstone monolith. The vault floor consists of clean concrete, that is, at the beginning of the simulation no contamination is contained in the floor. During a simulation the floor can have the affect of being a contaminant "sponge" thereby adding to the retardation of the contaminant transport.

1.2.3.5 Container *WallFloor*

The *WallFloor* container is in the model to account for the manner in which it was modeled in PorFlow. This will be discussed in detail in the benchmarking report (Taylor, 2009). Briefly, in PorFlow the wall sits on the floor which allowed for a separate flow path and perhaps a different material depending on vault type. This modeling caused the wall flow to be diverted around the floor into the dirt contiguous to the floor, flow around the floor, and then move under the vault beneath the floor due to the higher suction in the drier dirt. The *WallFloor* container's elements provided for a means by which these phenomena could be simulated by having both concrete and dirt flow paths.

1.2.3.6 Bypass Elements

One can see that there are several elements names containing "Bypass". These are used for Case C where there is assume to be a circumferential crack isolating the grout from the vault wall. When this is the case, there can be no diffusion between the grout and the other two flowpaths but there can be diffusion between the saltstone grout and the flowing medium in the crack. This is simulated by assuming the travel time in the crack is insignificant so that the diffusive flux term is applied directly to the unsaturated zone. The modeling assumption in PorFlow is that the crack extends through the vault floor,

therefore, the preceding assumption is correct. The *BypassDiffusionRate* element will be discussed in Section 1.5.

1.2.3.7 Container *Diffusion*

This container is the local implementation of the analytical diffusion model described in Section 1.5. Each vault type has its own implementation, but because of the linear modeling paradigm implemented for Vaults disposal cells, all Vault2's behave in an identical manner.

1.2.4 Container *UnsatZone*

UnsatZone (Figure 11) represents the flow field between the bottom of the vault floor and the top of the saturated zone. Consistent with the PorFlow modeling, a representative height is used for this flow zone. The height is a stochastic element so that the affect of varying distance to the water table may be assessed. Note that the height will vary for all Vaullt2 disposal cells in an identical manner. Nodalization studies performed for the FTF PA showed that the number of mixing cells used for this model should adequately model the unsaturated zone for any reasonable heights. The porous medium is a mixture of clayey soil and sandy. Benchmarking showed the *ClayeySoiFrac*[tion] to be zero.

The inlet flowpaths to *UZCell_In* are *VF5* (vault floor), *WFD05* (wall floor dirt), and *VW20* (vault wall). The outlet flow is evenly divided among the *WasteFootprint* cells.

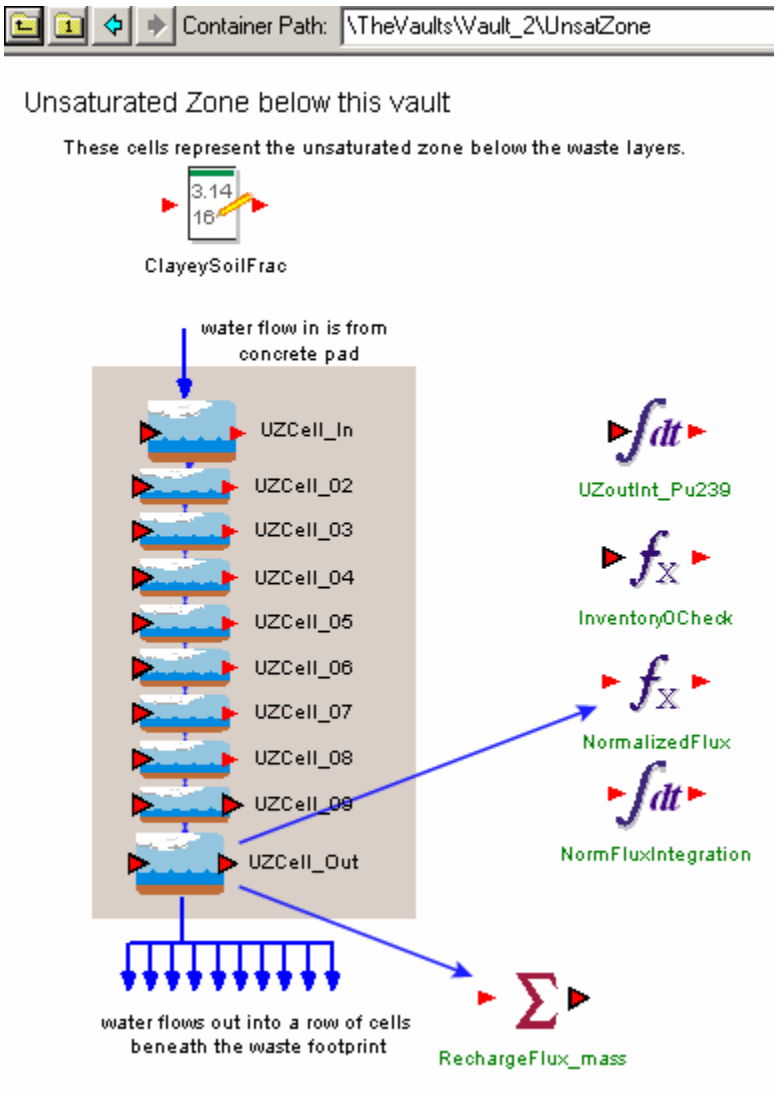

Figure 11 Container *UnsatZone*

1.2.5 Saturated Zone Modeling

The saturated zone consists of two containers, *WasteFootprint* and *NearWell*. *WasteFootprint* is the section of the saturated zone directly under the waste zone. *NearWell* is the 100 m boundary.

1.2.5.1 Container *WasteFootPrint*

This element contains 10 mixing cells. Its purpose is to account for the time delay of the upstream waste leaving the shadow of the waste form. This proved to be important in the FTF PA, and as the SDF model was being developed it was not known if this would be an important phenomenon here or not. It turns out that it is not important because the flow streamlines tend to be perpendicular to the long dimensions of Vault1 and Vault4

but the model structure was not changed once this became known. The porous medium is saturated sandy soil.

1.2.5.2 Container *NearWell*

NearWell represents the distance from the downstream edge of the waste form's shadow to the point of interest, the 100 m boundary. It consists of a number of mixing cells. Previous modeling efforts (e.g., the FTF PA analysis) showed a fairly fine discretization is necessary in order to avoid undue numerical dispersion.

This analysis will be quite different than previous SRS Goldsim models because of several factors. First, only one Vault 2 is being modeled so somehow each individual cell's distance to the point of interest must be determined. These distances vary from 100 m to about 550 m. Second, the saturated zone flow rates are not as uniform as they were in FTF or E-Area. Those analyses used a single value for the saturated zone flow rate. The implications of these differences will be discussed below.

As in previous SRS waste transport models, a plume function will be used to account for plume overlap. Its implementation in this model is discussed in Section 1.6. The use of the plume function in conjunction with the factors mentioned above leads one to model in the following manner.

When using a plume function, one could run an arbitrarily small flow area model, thereby causing the calculated concentrations to be higher than expected. The plume function is then used as a correction to that arbitrary concentration by analytically solving a dispersion equation. The reason a transport mechanism is needed it to have the correct timing of events. One could think of it as determining the arrival of a breakthrough curve. In our case, the timing is important because of radioactive decay. PorFlow is taken as the benchmark, so one can obtain timing information from it. All that is then necessary is to determine where along the GoldSim cell network the timing agrees with the PorFlow data. The GoldSim concentration at that point then can be used as the input to the plume function which will then supply a concentration to be used in the dose calculation.

What this is saying is that all one needs is a number mixing cells which will represent different places in time. The spatial aspect is removed so that one set of cells can be used to represent all the Vault 2s.

1.3 Inventory

Two inventory classes are used in this model. They are described in the following sections.

1.3.1 Disposal Cells Inventory

The *Inventory* container (Figure 12) is where the disposal cells inventory is defined. *Vault_Inventories* contains a unit curie for each of the vault types. [South,North]VaultSourceMultipliers is where the fractional curie is applied to each disposal cell for each radionuclide. The evolution of these terms is rather circuitous. The uncertainties applied to the inventories are based on a plus/minus fraction of the nominal inventory. Therefore, within each of the Multiplier containers is an element *Nominal[North, South]SourceInventory*. These values are used in Container *InventoryUncertainty*. This container contains three containers, one for each of the vault's inventory uncertainty. The uncertainty values are passed back to the Multipliers elements and it is that value that is passed to the dose calculation.

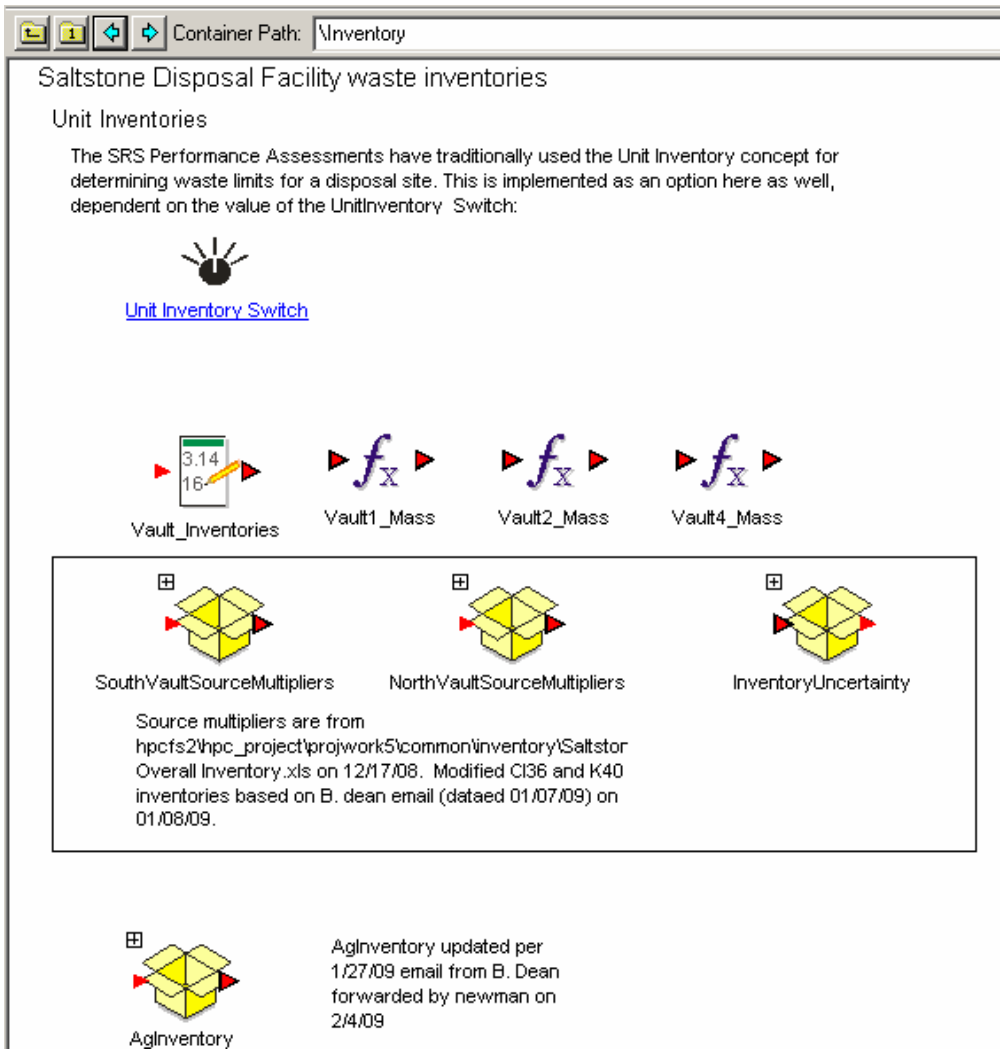

Figure 12 Container Inventory

1.3.1.1 Disposal Cell Inventory Uncertainty

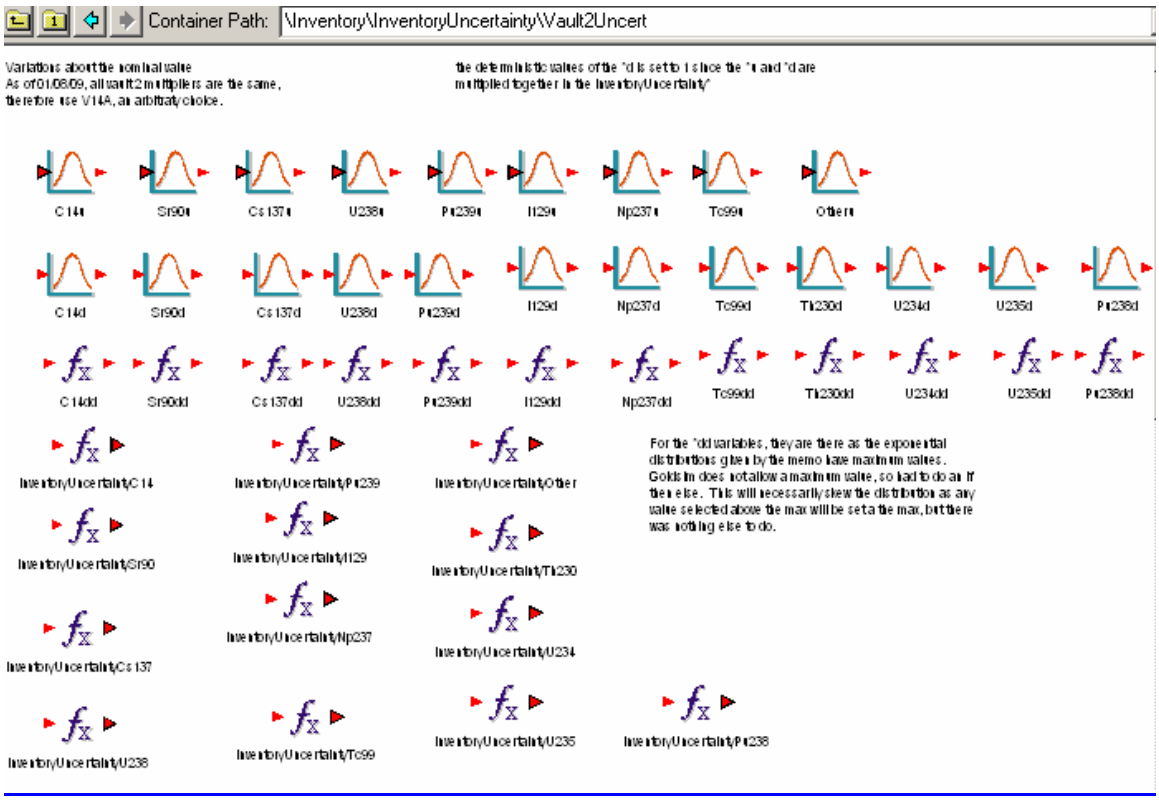

Figure 13 shows an example of the inventory uncertainties. Two uncertainties are applied to future disposal cells and one to the extant cells. The uncertainty applied to all disposal cells is the analytical (sampling, etc.) uncertainty. This is denoted by *u. Because it is unknown which future cells will get residual waste from which tank, this is an uncertainty to be applied to future cells only. These will be denoted as *d. Note that a decisions was made not to conserve curies. Therefore, because the same uncertainty factor will be applied to all cells, there could be little or much more inventory than the expected amount. For those radionuclide uncertainties which have a Goldsim element, an uncertainty was determined specifically for that radionuclide as it was determined to be one of interest. For all radionuclides not specifically called out, a single general uncertainty value is applied and is denoted by *InventoryUncertaintyOther*.

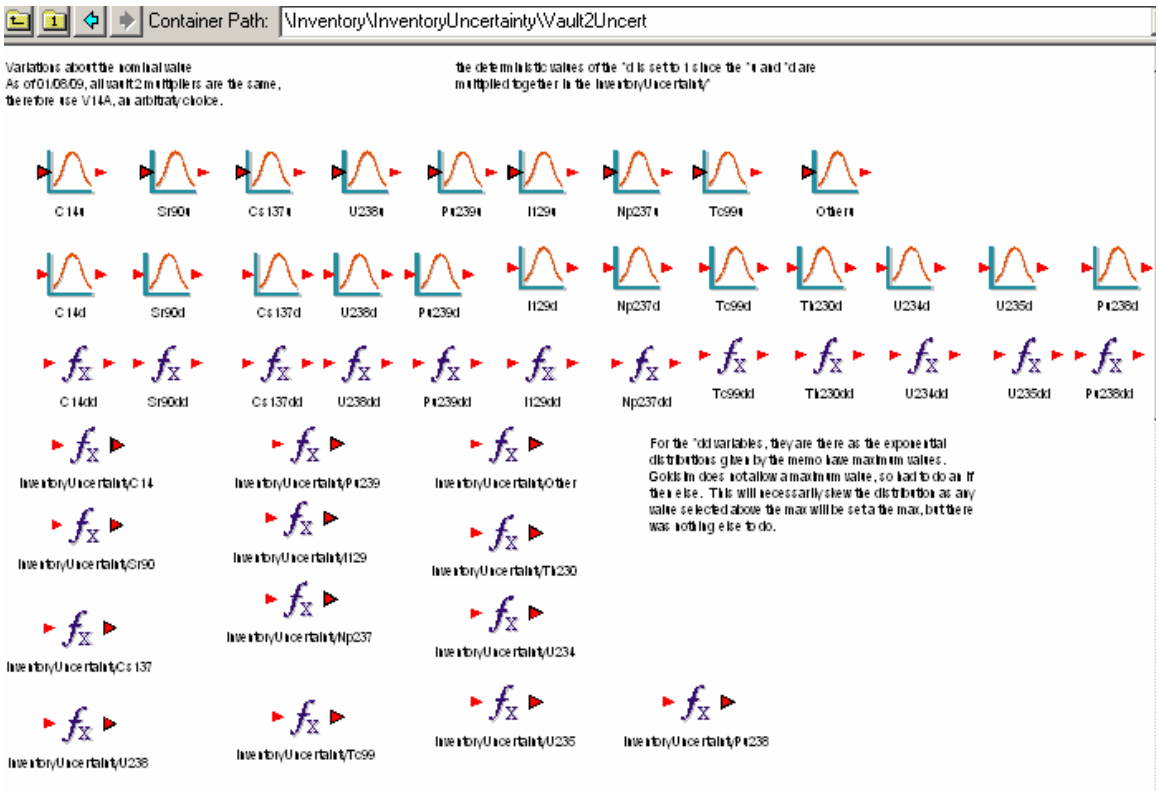

Figure 13 Inventory Uncertainty Example

1.3.2 Post-drilling Inventory

This inventory, *AgInventory*, is available for use in the Inadvertent Human Intruder analyses. It represents the drill cuttings inventory if one were to drill into a disposal cell. For the SDF analysis this eventuality is considered to be a sensitivity. The base case is defined as no drilling into a disposal cell.

1.4 Flow Abstraction

This section is a discussion of the algorithm used to extract the 2-dimensional flow field from the PorFlow unsaturated zone analysis into the GoldSim 1-dimensional advective model. GoldSim does not calculate flow fields, it uses them as boundary conditions. PorFlow generates its time dependent flow files by running 44 independent, steady-state flow runs, with each of the 44 representing a period in time. Each of those flow runs generates an output file which contains the Darcy velocities for each computational cell. The coordinates are either x-y for the rectangular disposal cells (Vaults 1 and 4), or r-z for the FDCs. Regardless, the output format is the same (except that the r-z is flipped from the x-y).

A VBA script was written to interrogate the output files and to summarize the results in an Excel workbook. For the waste and dirt zones, the files were interrogated at 20

different elevations, corresponding to the 20 mixing cells used to represent each, and ten radial locations for the grout and four for the dirt. An average was then taken for each elevation. A table was then constructed of time period versus elevation. Figure 14 shows a small section of one of these tables. "Time Periods" corresponds to the PorFlow time periods and "i=37", etc. corresponds to a PorFlow elevation index. "TI44" is the last time period. If one were to read across a row, one would note that the velocities are essentially same, as one would expect for the grout for Case A.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Time Period	i=37	i=40	i=43	i=44	i=45	i=47	i=48	i=49	i=51	i=52	i=53
2	TI44	-0.26922	-0.25867	-0.24622	-0.24389	-0.242	-0.23967	-0.23889	-0.23833	-0.23789	-0.23778	-0.23744
3	TI43	-0.25778	-0.24933	-0.23944	-0.23767	-0.23667	-0.23489	-0.23444	-0.23378	-0.23344	-0.23356	-0.23356
4	TI42	-0.10443	-0.10753	-0.11247	-0.11392	-0.11551	-0.11794	-0.11877	-0.11956	-0.12078	-0.12122	-0.12189
5	TI41	-0.09174	-0.09391	-0.09693	-0.09793	-0.09888	-0.10078	-0.10158	-0.1022	-0.10322	-0.10361	-0.10401
6	TI40	-0.08559	-0.08754	-0.09022	-0.09111	-0.092	-0.09363	-0.09441	-0.09499	-0.09584	-0.09622	-0.09656
7	TI39	-0.08266	-0.08454	-0.08707	-0.08791	-0.08874	-0.09036	-0.09103	-0.0916	-0.09253	-0.09279	-0.0931
8	TI38	-0.0688	-0.07046	-0.07254	-0.07319	-0.07384	-0.07498	-0.07539	-0.07577	-0.07621	-0.07632	-0.07639
9	TI37	-0.0674	-0.06901	-0.07103	-0.07164	-0.07229	-0.07341	-0.07382	-0.07418	-0.0746	-0.0747	-0.07478
10	TI36	-0.06601	-0.06757	-0.06953	-0.07016	-0.07074	-0.07184	-0.07226	-0.07257	-0.07299	-0.07311	-0.07317
11	TI35	-0.0646	-0.06613	-0.06803	-0.06867	-0.06921	-0.07029	-0.0707	-0.07101	-0.07141	-0.07156	-0.07158
12	TI34	-0.0632	-0.06471	-0.06653	-0.06712	-0.06768	-0.06871	-0.06912	-0.06944	-0.06982	-0.06993	-0.06997
13	TI33	-0.06178	-0.06324	-0.06507	-0.06562	-0.06619	-0.06719	-0.06756	-0.06789	-0.06826	-0.06837	-0.06839
14	TI32	-0.06038	-0.06181	-0.06358	-0.06413	-0.06469	-0.06562	-0.06601	-0.06631	-0.0667	-0.06679	-0.0668
15	TI31	-0.05897	-0.06034	-0.06207	-0.06263	-0.06312	-0.06408	-0.06443	-0.06472	-0.0651	-0.06518	-0.06523

Figure 14 Sample Velocity Table – Grout

Figure 15 shows a similar table for the dirt zone, with several columns left out. "i=37" is the innermost node and "i=73" the outermost. The total horizontal distance is about 20 feet, so one can see the steep flow gradient in this zone.

Time Perid	i=37	i=40	i=43	i=44	i=45	i=	i=62	i=64	i=66	i=69	i=73
TI44	-103.467	-105.633	-107.133	-107.933	-108.367		-127.067	-130.3	-134.2	-140.433	-147.133
TI43	-112.2	-106	-106.867	-106.967	-107.733		-126.933	-131.133	-136	-144.533	-153.867
TI42	-114.2	-114.667	-116.9	-117.7	-118.767		-155.133	-162.667	-171	-186.267	-204.433
TI41	-114.2	-114.667	-116.9	-117.7	-118.8		-155.133	-162.667	-171.333	-186.633	-204.767
TI40	-114.2	-114.667	-116.9	-117.7	-118.8		-155.133	-162.667	-171.333	-186.633	-204.767
TI39	-114.2	-114.667	-116.9	-117.7	-118.8		-155.133	-162.667	-171.333	-186.633	-204.767
TI38	-72.7333	-72.6333	-72.6667	-72.7	-72.7333		-73.2667	-73.2667	-73.2667	-73.2	-73.1
TI37	-72.4667	-72.4	-72.4667	-72.5	-72.5333		73	72.9667	72.9667	72.8667	72.7667

Figure 15 Sample Velocity Table – Dirt

Examination of the flow profiles in the vault walls showed them to be essentially uniform. Therefore, one value was used to represent each of the time periods for vault wall flow.

1.4.1 Flow Abstraction Program Description

The VBA script used to abstract the flow profiles used in the Goldsim model is shown in Appendix B. The program is written in Visual Basic for Applications and is linked to Excel workbooks. Because of the differences in PorFlow noding and other PorFlow idiosyncrasies to be discussed later, each vault type has its own version of the program.

The example in Appendix B is for Vault2. Brief descriptions of the subprograms which abstract the PorFlow flow data are described next.

1.4.1.1 Subroutine driver()

The subroutine driver() runs through all the subroutines to generate the flows used by the GoldSim model. Perhaps most importantly, it writes the names of the PorFlow files from which the data were extracted into the workbook.

1.4.1.2 Subroutine GetPorflowFlows(filesfrom)

This purpose of the GetPorflowFlows(filesfrom) subroutine is to gather the PorFlow flow data into a single Excel workbook. The data are in 44 separate folders, each representing a steady-state flow period. The subroutine loops through all 44 folders to extract the downward flow component, which for Vault2 is in the z-direction, and for Vaults 1 and 4 in the y-direction. This is important in that the PorFlow output file which is interrogated to extract the flow information always references its output in terms of (i,j). For radial geometry (Vault2) the correspondence is (z,r). For rectilinear geometry the correspondence is (x,y).

The subroutine reads the PorFlow output to find the beginning of the downward flow information. The output file is essentially a text file. As such the subroutine reads it line-by-line looking for key words. The subroutine split_on_blanks is called because the VBA inline function treats each blank character as a separate character when the information desired is only non-blank characters. When the correct data are found they are written to the appropriate worksheet in the workbook. By doing this one has a local summary of the trenchant data.

1.4.1.3 Subroutine summarytables()

The data extracted by GetPorflowFlows represents the entire flow field of the PorFlow model. The GoldSim model needs data for 4 flow regions: saltstone, vault wall, dirt contiguous to the wall, and the unsaturated zone beneath the vault. The summarytables() subroutine selects the appropriate data for each flow region and writes it to the appropriate worksheet. The saltstone and dirt cells have a 1-to-1 correspondence and are treated similarly.

The PorFlow model's vertical axis corresponds to the GoldSim model's 1-dimension. To abstract in radial (r- or x-) dimension a spatial averaging is done for each of the twenty cells in the GoldSim model's vertical stack. Ten locations in a horizontal plane are sampled for the saltstone and four for the grout. These two sample groups are then

individually arithmetically averaged and that average value is the value used for that mixing cells flow during that time period.

The unsaturated zone's flow gradient in the vertical direction is fairly uniform. Four horizontal planes are chosen with 16 points being sampled in each plane. The unsaturated zone "column" of interest is that which lies beneath the disposal vault and the dirt contiguous to the vault. The horizontal dimension is the sum of the saltstone and dirt distance, hence the more samples.

One drawback to this method is that some of the Cases can have a very steep flow gradient at one of the region boundaries, the flow can vary over several orders of magnitude in just a few PorFlow computational cells. This is particularly true in the saltstone. These gradients can skew the average. While this may not be aesthetically pleasing, the end affect on dose was considered acceptable because the wall flux is what drives the dose during the period of interest.

Flow in the vault wall is essentially constant in both the vertical and horizontal dimensions so long as one stays away from the edges where the PorFlow differencing scheme can provide some gradient. Therefore, one sample from basically in the middle of the wall, in both dimensions, is used.

1.4.2 Subroutines FlipGrout(), FlipUZ(), FlipDirt()

The FlipGrout(), FlipUZ(), and FlipDirt() routines are used to set the abstracted data in a form which can be used directly by the GoldSim model. The Vault2 model requires a flip in both time and space, while Vaults 1 and 4 needed only to be flipped in time. These routines also convert the flow from negative (the PorFlow convention) to positive (the GoldSim convention).

1.5 Diffusion Model

Porflow analyses showed diffusion to be an important mechanism for the transport of contaminant from the wasteform to the external environment. Because the saltstone grout is relatively durable over time, the major contributor to contaminants at the compliance points is the vault wall. The mechanism by which contaminants get to the vault wall outside environment is diffusion. This section will discuss the implementation of an analytical solution to diffusive mass transfer. A detailed explanation of the solution can be found in [Flach, 2009].

The Goldsim diffusion methodology was first used in order to assess its efficacy. Nodalization studies showed than for a reasonably well converged solution a 2-dimensional cell structure of over 400 coupled mixing cells would be required. This would have had a very adverse affect on the run-time of the model so an alternative approach was needed.

The alternative approach used was analogous to that developed by the Porflow modelers to determine sulfate penetration into the concrete and its concomitant affect on ^{99}Tc release. The model is a shrinking core model which accounts for the concentration gradient as function of time. The shrinking core model required by the Goldsim fate and transport model is somewhat more complicated than the shrinking core model required by the Porflow analysis in that the Goldsim has three components which must be considered: the saltstone, the wall, and the dirt.

1.5.1 Diffusion Model Implementation

The diffusion model is implemented for each of the three vault types. The implementation for Vault2 is shown in Figure 16. One thing to keep in mind is that the solution is valid only for those radionuclides which exist at t=0, that is, the model does not explicitly recognize in-growth. However, in-growth is implicit to the model via the species concentrations used as arguments in the model. This is not seen to be a major drawback. The benchmarking process allowed for the model to be fine-tuned such that its behavior is quite similar to the fully coupled diffusion model implemented by Porflow.

The Goldsim implementation of the model has two purposes, 1) to simulate the mass transfer into the dirt outside of the vault and 2) the build-up, or decrease, of the contaminant mass in the vault wall. As described in [Flach, 2009], the dirt is seen as an infinite sink, that is, there is no increasing concentration in the mixing cells making up the dirt column. This is a valid assumption because the flowrate in the dirt is great enough so that all contaminants (not sorbed onto solids) which diffuse during one time step are advected out of that cell during the timestep.

The implementation of the equations is straight forward and will not be discussed. Several idiosyncrasies of Goldsim make the implementation more challenging. The idea behind the implementation is that the analytical solution will dictate the amount of mass transported by diffusion. The mass transfer will be accomplished by the use of a direct transfer pathway. However, this requires a fractional rate of transfer. This requires checks so that a negative transfer can not occur, no more mass than is in the mixing cell can be transferred, and that there be a non-zero mass so that there is not a divide by zero when computing the fraction to be transferred. These requirements lead to much of the manipulation required.

Goldsim sees a recursive loop if one wishes to base the mass transfer rate on the current time step's mass. The use of a *PreviousValue* element is required. Due to the rate of change and the size of the time steps the use of a previous time step value does not significantly affect the results.

1.5.2 Diffusion Model Implementation for Vault 2 Case C

Vault 2 Case C is a special case for the implementation of the diffusion solution. Case C assumes that a complete, circumferential crack occurs between the saltstone and the vault wall and that the crack continues through the floor of the vault. This crack constitutes a fast flow path. This fast flow has a vastly different affect than the one encountered in the FTF PA GoldSim modelling. In FTF model, the crack diverted infiltration from the grout to the waste which was a thin layer on the bottom of the tank. This caused all the waste to be contacted by an increased flow. The crack in the Saltstone model vaults diverts flow from the saltstone, which in this case is the waste form. Therefore, although the infiltration rate is greater, it is likely that less waste will be affected by the flow.

The circumferential crack isolates the saltstone from the rest of system in the radial direction. Therefore, the only diffusive path is that from the saltstone to the crack. The high flowrate in the crack makes it appear as an infinite sink to the salstone's diffusive flux. This is analogous to the appearance of the dirt in the typical case. The analytical solution to be applied is the one described as the "long term" solution in [Flach, 2009]. Because of the assumption that the circumferential crack extends through the floor of the vault, and the travel time in the crack is insignificant compared to the time steps, the direct mass transfer pathway is from the saltstone to the unsaturated zone inlet mixing cell.

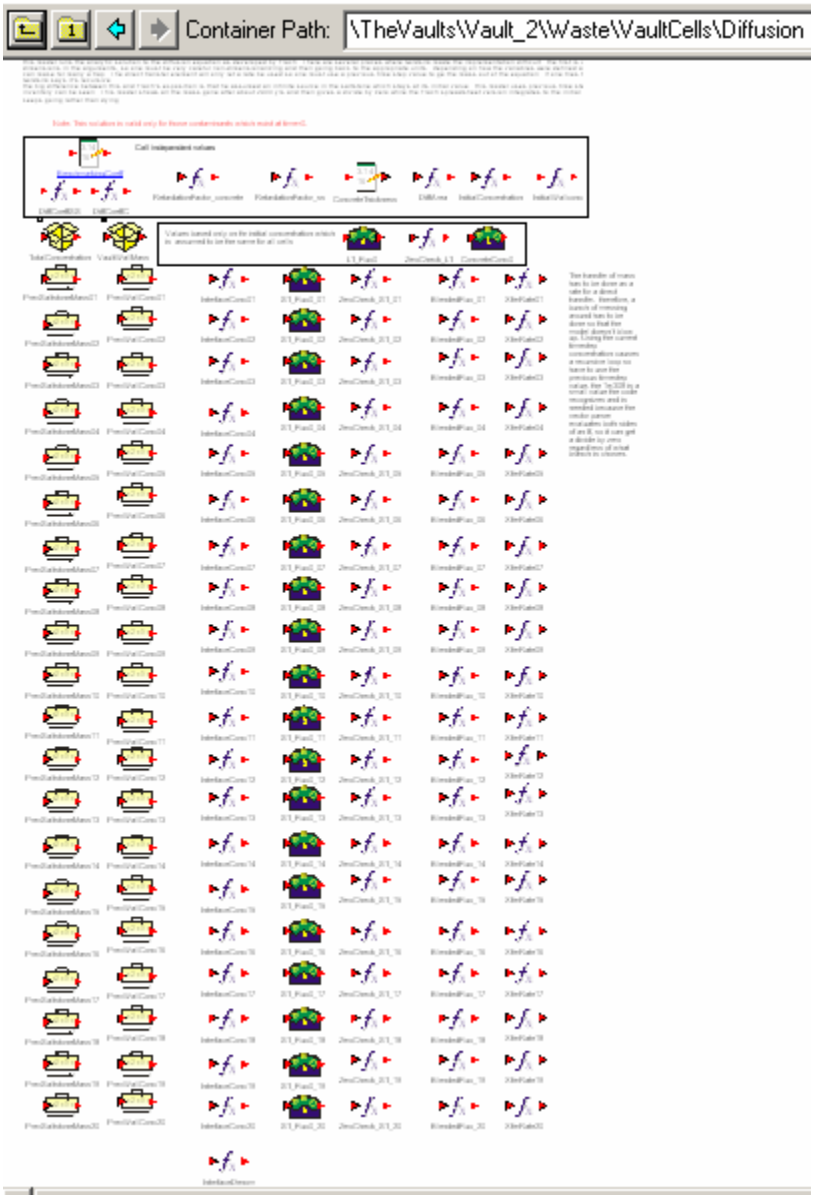

Figure 16 Diffusion Model Implementation

1.6 Plume Model

1.6.1 Plume Model Implementation

The plume model is implemented for all saltstone disposal cells to account for plume overlap. Using the 12 flow sectors defined by PorFlow (see Figure 18) and 64 disposal vaults, a matrix of 64 x 12, or 678 elements, is created. Many of those elements will have no affect on the superposition of the plumes, and although those elements cannot be

eliminated from the matrix, their affect can be neglected. The logic used for setting up the plume overlap calculation is presented below.

The plume function is designed to calculate a factor to account for transverse dispersion of the contaminant. (See Appendix B for the denouement of the plume function.) In order for this to happen, a reference point must be chosen. The point chosen for this analysis is the center of the PorFlow calculated plume, as evinced by the streamlines shown in Figure 18. This is fairly obvious. Less obvious is how to compare, as one inevitably must during calibration, to the PorFlow generated concentration data. The data to be supplied is the maximum value in any computational cell in any aquifer in a sector. The data can move during a calculation which makes the calibration even more problematic. The decision was made to use the center node of each sector as the sectors' reference points. Therefore, the plume function is calculating the spread of a plume from where a disposal cell's streamline crosses the 100 m boundary to the center of a sector at the 100 m boundary.

Values for plume overlap are calculated only for plumes which may reasonably be thought to interact. There is no plume interaction between the north and south section of the SDF, nor are there any interactions between plumes on opposite sides of the flow divide. This brings up a fine distinction in what is meant by plume overlap. In this report it is used to represent the case where contaminants from multiple sources feed the same point in space. The Porflow model has three aquifers, the lowest of which is essentially isolated from the top two.

Figure 17 shows that the majority of the contaminant from Vault 4 flows in an eastwardly direction. However, there is a small contribution in sectors to the north. While the Vault 4 contribution of some contaminants may be small in absolute terms, relative to the contributions from the Vault 2's the contribution may be significant. The northern flow occurs in the lowest, isolated aquifer. To be consistent with how Porflow passes its data to the dose calculator the contribution to the mixing cell concentrations was accommodated during the calibration process.

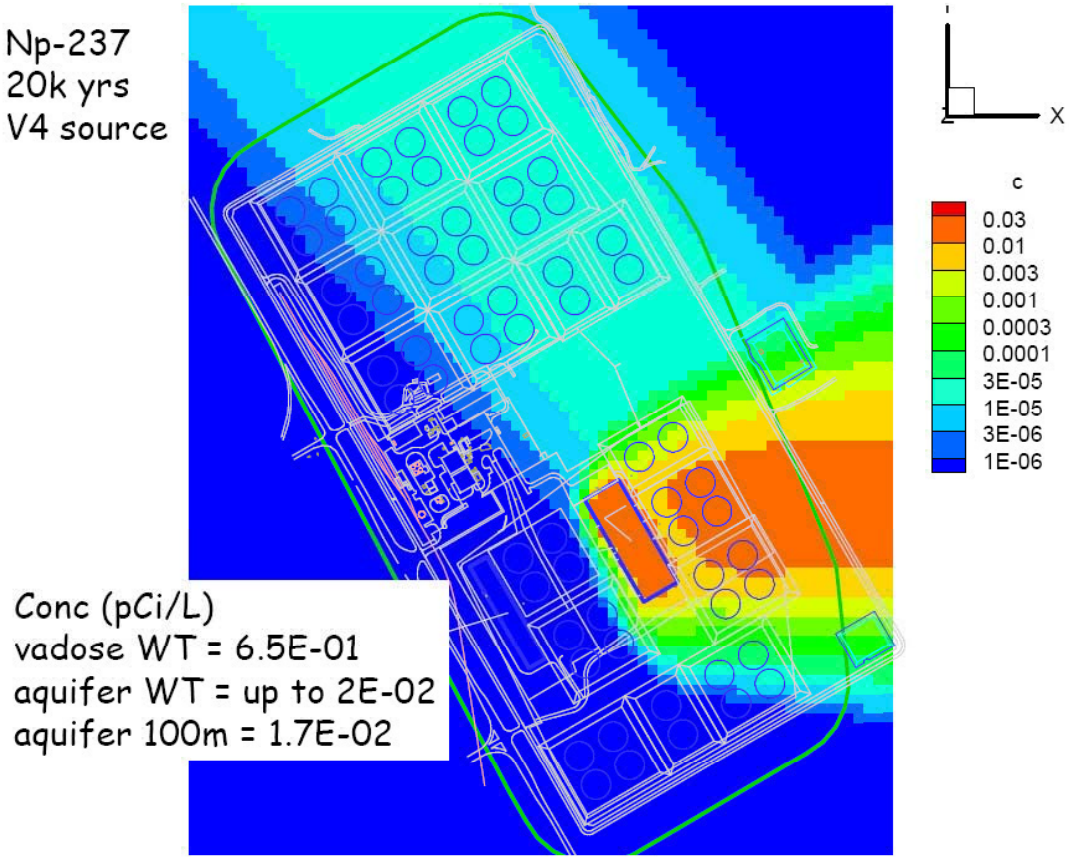

Figure 17 Vault 4 Plumes

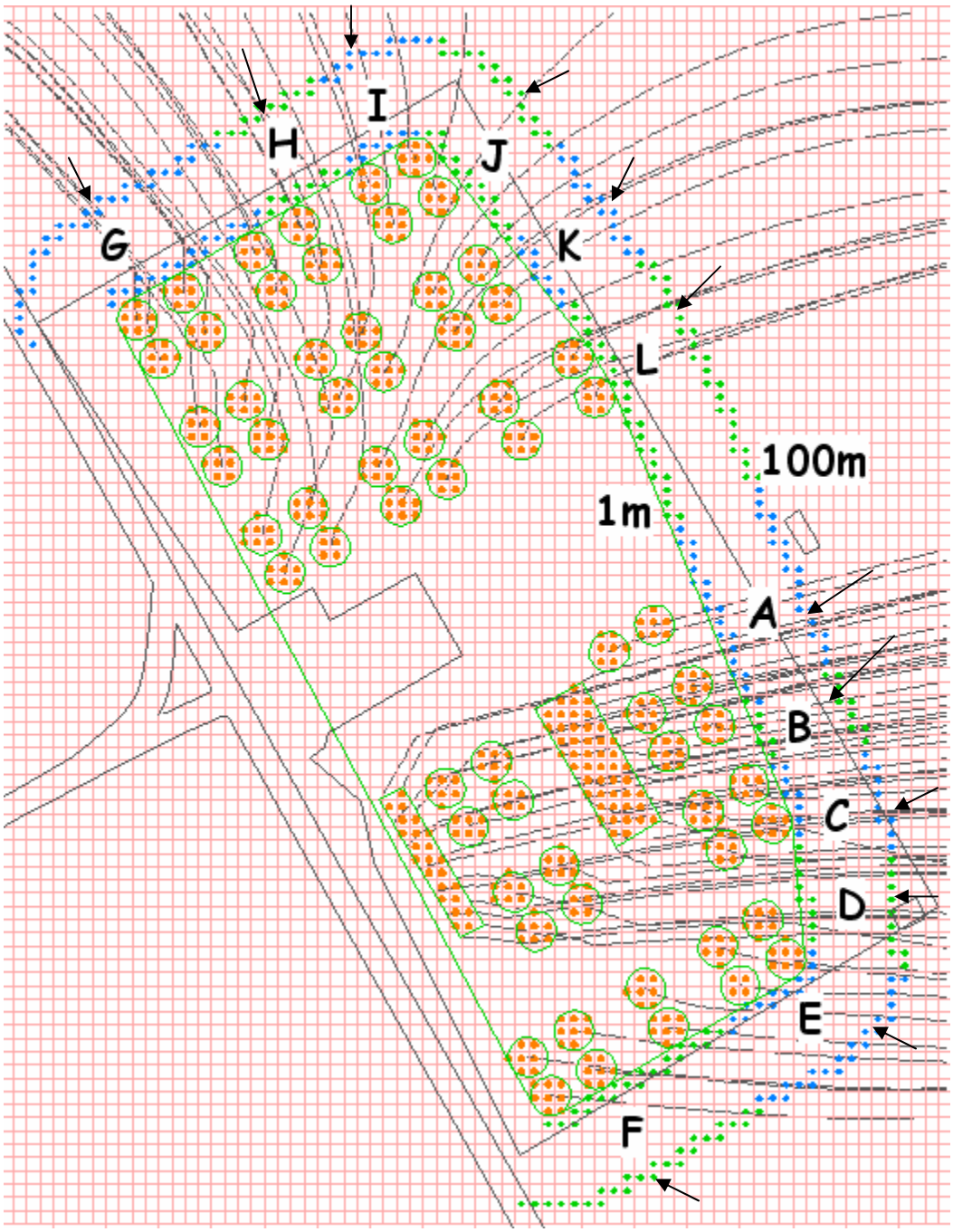

Figure 18 SDF Sectors

Table 1 shows the plume offset matrix with the values being distance in feet. A distance of 0 ft implies that the streamline passes over the center node of a sector. A distance of 1e20 ft implies the point is infinitely far and has infinite dilution, therefore no contribution to the overlap. The bold numbers in the table are those with a plume overlap contribution. The arrows in Figure 18 show the location of the reference cells for each sector.

Table 1 Plume Offsets

	South Section						North Section					
	A	B	C	D	E	F	G	H	I	J	K	L
V2A	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**4.80E+02**	**0.00E+00**
V2B	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**6.40E+02**	**1.60E+02**
V5A	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**8.00E+01**	**8.80E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V5B	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**1.60E+02**	**5.60E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V5C	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**8.00E+01**	**8.80E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V5D	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**1.60E+02**	**5.60E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V6A	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**4.80E+02**	**2.80E+02**	**7.20E+02**	1.00E+20	1.00E+20	1.00E+20
V6B	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**6.40E+02**	**8.00E+01**	**1.48E+03**	1.00E+20	1.00E+20	1.00E+20
V6C	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**4.80E+02**	**2.80E+02**	**7.20E+02**	1.00E+20	1.00E+20	1.00E+20
V6D	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**6.40E+02**	**8.00E+01**	**1.48E+03**	1.00E+20	1.00E+20	1.00E+20
V7A	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**3.20E+02**	**1.20E+02**	1.00E+20	1.00E+20
V7B	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**5.84E+02**	**1.60E+02**	1.00E+20	1.00E+20
V7C	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**4.00E+02**	**0.00E+00**	1.00E+20	1.00E+20
V7D	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**4.80E+02**	**3.20E+02**	1.00E+20	1.00E+20
B8A	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**4.00E+01**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V8B	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**2.00E+02**	**5.60E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V8C	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**4.00E+01**	**8.00E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V8D	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**2.40E+02**	**5.60E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V9A	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**7.20E+02**	**8.00E+01**	**5.20E+02**	1.00E+20	1.00E+20	1.00E+20
V9B	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**3.20E+02**	**3.60E+02**	1.00E+20	1.00E+20	1.00E+20
V9C	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**0.00E+00**	**5.20E+02**	1.00E+20	1.00E+20	1.00E+20
V9D	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**1.20E+02**	**3.20E+02**	1.00E+20	1.00E+20	1.00E+20
V10A	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**2.40E+02**	**2.80E+02**	1.00E+20
V10B	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**4.37E+02**	**1.20E+02**	1.00E+20
V10C	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**4.80E+02**	**0.00E+00**	1.00E+20
V10D	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**1.20E+02**	**4.00E+02**

	South Section						North Section					
	A	B	C	D	E	F	G	H	I	J	K	L
V11A	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**2.00E+02**	**5.60E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V11B	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**4.80E+02**	**2.64E+02**	**8.40E+02**	1.00E+20	1.00E+20	1.00E+20
V11C	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**4.80E+02**	**2.64E+02**	**7.20E+02**	1.00E+20	1.00E+20	1.00E+20
V11D	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**0.00E+00**	**4.40E+02**	1.00E+20	1.00E+20	1.00E+20
V12A	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**5.20E+02**	**0.00E+00**	**5.20E+02**
V12B	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**3.20E+02**	**1.60E+02**
V12C	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**5.20E+02**	**0.00E+00**	**5.20E+02**
V12D	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**1.60E+02**
V13A	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**5.20E+02**	**0.00E+00**
V13B	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**2.00E+02**
V14A	**2.00E+02**	**3.20E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V14B	**2.00E+02**	**3.20E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V14C	**5.60E+02**	**8.00E+01**	**2.40E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V14D	**4.80E+02**	**0.00E+00**	**3.20E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V15A	**1.20E+02**	**6.40E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V15B	**1.60E+02**	**7.20E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V16A	**2.80E+02**	**2.80E+02**	**5.60E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V16B	**1.60E+02**	**3.20E+02**	**6.40E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V16C	**5.20E+02**	**0.00E+00**	**3.20E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V16D	**4.00E+02**	**1.20E+02**	**4.00E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V17A	1.00E+20	1.00E+20	**2.80E+02**	**1.20E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V17B	1.00E+20	1.00E+20	**2.80E+02**	**1.20E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V17C	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**6.00E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V17D	1.00E+20	1.00E+20	1.00E+20	**1.20E+02**	**3.60E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V18A	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V18B	1.00E+20	**4.00E+02**	**4.00E+01**	**3.20E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V18C	1.00E+20	**6.00E+02**	**2.40E+02**	**1.20E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V18D	1.00E+20	**5.60E+02**	**2.80E+02**	**1.60E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V19A	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**1.60E+02**	**8.00E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V19B	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**8.00E+01**	**1.00E+03**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20

	South Section						North Section					
	A	B	C	D	E	F	G	H	I	J	K	L
V19C	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**4.80E+02**	**4.80E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V19D	1.00E+20	1.00E+20	1.00E+20	1.00E+20	**1.60E+02**	**8.00E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V20A	1.00E+20	1.00E+20	1.00E+20	**8.40E+02**	**3.20E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V20B	1.00E+20	1.00E+20	1.00E+20	**7.20E+02**	**1.60E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V21A	1.00E+20	1.00E+20	1.00E+20	**3.60E+02**	**4.80E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V21B	1.00E+20	1.00E+20	1.00E+20	**2.00E+02**	**6.00E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V21C	1.00E+20	1.00E+20	1.00E+20	**4.80E+02**	**4.80E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V21D	1.00E+20	1.00E+20	1.00E+20	**4.00E+02**	**4.80E+02**	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V1	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20
V4	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20	1.00E+20

1.7 Dose Model

This document will discuss the dose model's implementation only. Discussions of the pathways, etc. can be found in [FTF PA, 2007]

1.7.1 Contaminant Concentrations

The dose model gets its contaminant concentrations from the *NearWell* elements of the vault containers. The transport portion of the model is run assuming a unit curie of each parent radionuclide. It is in the determination of the well concentration where the actual inventory is applied. One must remember that because the model is based on distribution coefficients, not solubilities, the concentrations scale linearly with initial inventory.

The concentrations are computed in container *ExposureMediaConc*. Container *SectorLocations* contains only graphical images. Container *WellCompletionDepth* is a holdover from the FTF model. It is not used in this analysis but is left in the model should later analyses wish to use it. *[North,South]VaultIndex* are a useful bit of Goldsim arcana. Any vector can be ordered in any way the modeler desires. If the order is changed it usually creates no issues. However, certain Goldsim elements, such as *GetItem* require the index, not the vector element name. This has been brought up to the Goldsim developers as the code has to do a reverse lookup to find the vector element, so why not be able to use the vector elements name? Regardless, the solution to this is to define these *VaultIndex*-type elements. An example of this use is shown below.

Containers *ByCellSectorConc* and *BySectorSectorConc* are different collections of the concentration data. *ByCellSectorConc* contains container *GetConcentrations*. In the latter container the concentrations are calculated in terms of g/L. The following example shows how the concentration is determined for disposal cell 8A's contribution to sector G.

```
Vault_2.Concentration_in_Water_2 * BenchmarkingFractionNorth[V8A,G] *
PlumeCalc_Sectors.PlumeCorrectionNorth[V8A,G]*getcolumn(NorthSourceMult,NorthVaultIndex[V8A])
```

The terms of the preceding expression, in order, are:

- The concentration from *Vault2*'s *NearWell*.
- Benchmarking fraction for disposal cell 8A's contribution to sector G. All benchmarking fractions are currently set to 0.5 although the construct is preserved if refinement is desired.
- The plume correction is as described in Section 1.6.1.

- The last term is where the inventory is applied. It also shows the application of the *VaultIndex* construct. The second argument of the *getcolumn* function requires a number. The *VaultIndex* construct allows the use of the disposal cell's name and can be used regardless of its position in the vector.

BySectorSectorConc is the summation of all disposal cells which contribute to a specific region. It supplies the values used in the dose calculation as the dose is reported by region.

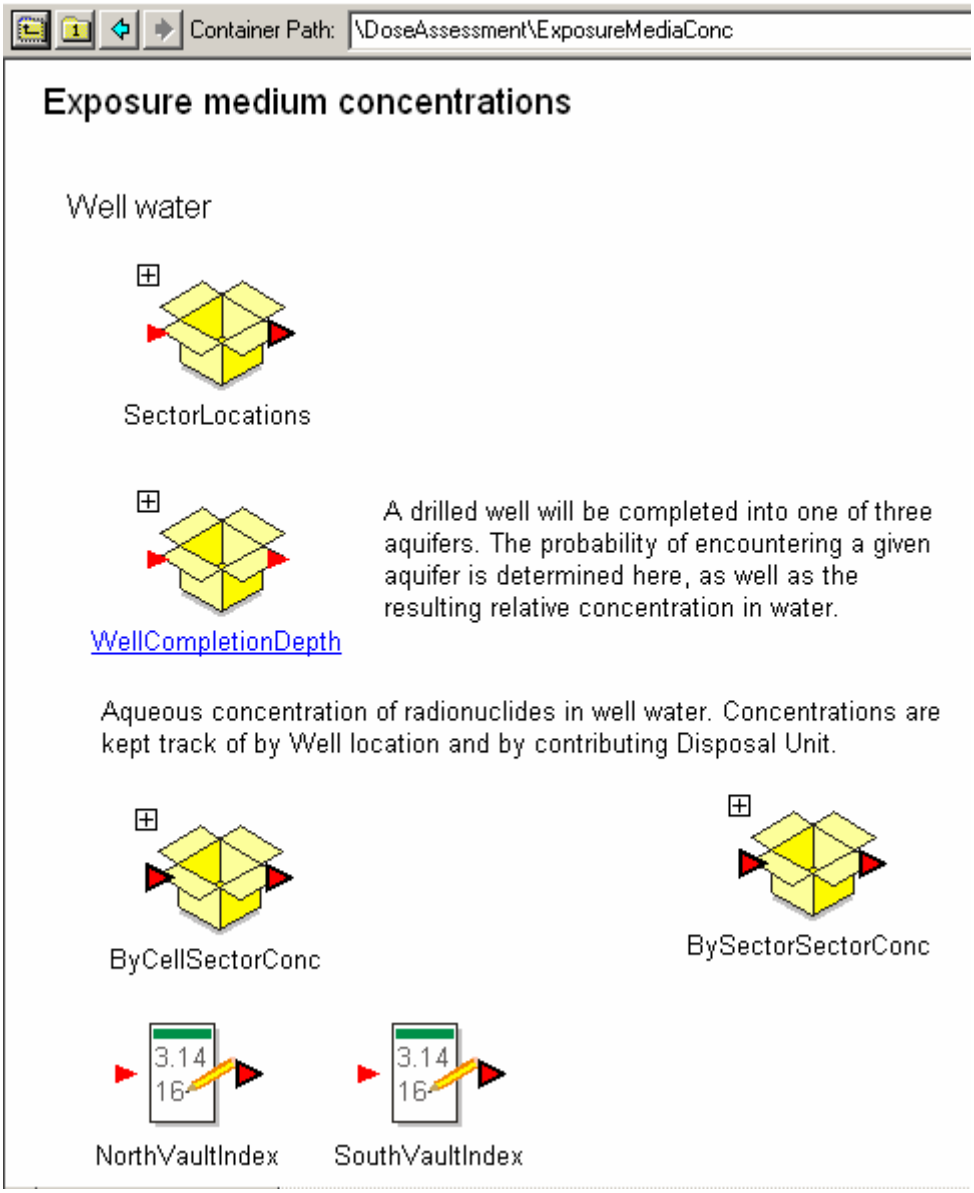

Figure 19 Container *ExposureMediaConc*

References

FTF PA, 2008

Taylor, Glenn, *Saltstone Disposition Facility Stochastic Transport and Fate Model Benchmarking*, SRNL-TR-2009-00052, March, 2009

Denham, Miles, *Estimation of Eh and pH Transitions in Pore Fluids During Aging of Saltstone and Vault 2 Concrete*, SRNL-TR-2008-00283, December, 2008.

Flach, G. P., *Approximate Solution for Diffusional Release from Saltstone Vaults*, SRNL-STI-2009-00114, Rev. 0, Feb. 2009.

Performance Assessment for F-Tank Farm at the Savannah River Site, SRS-REG-2007-00002, December, 2007.

Appendix A Flow abstraction program

Private Sub driver()

```
GetPorflowFlows filesfrom
summarytables
FlipGrout
FlipUZ
FlipDirt

Sheets("info").Cells(1, 1).Value = filesfrom & " where x: =
hpcfs2\hpc_project\projwork5\saltstone"
Sheets("info").Cells(2, 1).Value = Date
MsgBox "all done"
End Sub
```

Private Sub GetPorflowFlows(filesfrom)

```
' this subroutine get the porflow results and stick them in a sheet for each time period
'
Application.ScreenUpdating = False
Application.DisplayAlerts = False
Set fso = New FileSystemObject
Set ydir = New RegExp
ydir.Pattern = "\*-\*- U   - X-DIR"  ' plot and out files have flipped axes
Set itest = New RegExp
itest.Pattern = "I ="

Dim SubfReg, InLine
Set SubfReg = New RegExp
```

```
SubfReg.Pattern = "\d\d$"



Dim Type1, Type3, Type4
Dim TankType, RowIndex, ColumnIndex, GeometricMean, HarmonicMean, StartColumn
TankType = Array("TypeI", "TypeIII", "TYPEIV")

Open "D:\PA\Saltstone\Porflow flows\flowdebug.txt" For Output As #10

iindex = 3
StartColumn = 1
itype = 0
   ' from \\g-flach\TankPA\VadoseType*\Flow  where * is tank type
   casename = "CaseA"
   filesfrom = "x:\Vault2\Flow\" & casename
   Set Type1 = fso.GetFolder(filesfrom)



   ColumnIndex = StartColumn
   For Each subf In Type1.SubFolders
      RowIndex = 3
      StartColumn = 1
      Set xsheet = Sheets.Add
      Set xmatch = SubfReg.Execute(subf)  ' get hte number of the filename


      xsubf = "TI" & xmatch(0)


      For Each sheetname In ActiveWorkbook.Sheets ' if the sheet already exists, delete it
         If sheetname.Name = xsubf Then
            sheetname.Delete
         End If
      Next sheetname
```

```
xsheet.Name = xsubf
startrow = 1
iblock = 2  ' hardwire in 9 blocks, this is the counter
Close #1
Open subf & "\RUN.DOS" For Input As #1
'Print #10, subf
Do While Not EOF(1)

   Line Input #1, InLine
   'Print #10, InLine

   If ydir.Test(InLine) Then  ' found where the y velocity starts
        ' skip a line

        Line Input #1, InLine

      Do While Not EOF(1)


        If StartColumn = 1 Then  ' first time so write indices
           For ii = 1 To 92  ' hardwire in the block
              Line Input #1, InLine
              'Print #10, InLine
              split_on_blanks InLine, outarray
              icol = StartColumn

              For iarray = 0 To UBound(outarray)
                 xsheet.Cells(RowIndex, icol).Value = outarray(iarray)
                 icol = icol + 1
              Next iarray
```

```
        RowIndex = RowIndex + 1

      Next ii


      Line Input #1, InLine
      Line Input #1, InLine  ' get the i indices
      'print #10, InLine & "  getting indices"
      split_on_blanks InLine, outarray
      icol = StartColumn + 2
      For i = 2 To UBound(outarray)


        xsheet.Cells(RowIndex, icol).Value = outarray(i)
        icol = icol + 1
      Next i
      RowIndex = RowIndex + 1
      icol = StartColumn + 2
      Line Input #1, InLine  ' get coordinates
      split_on_blanks InLine, outarray

      For i = 2 To UBound(outarray)
        xsheet.Cells(RowIndex, icol).Value = outarray(i)
        icol = icol + 1
      Next i
      StartColumn = icol - 1
      Line Input #1, InLine
      Line Input #1, InLine

    'Print #10, InLine & "next block"
```

```
Else ' get the other blocks
   RowIndex = 3
   For ii = 1 To 92  ' hardwire in the block

      Line Input #1, InLine
      split_on_blanks InLine, outarray
      icol = StartColumn

      For iarray = 2 To UBound(outarray)
          'Print #10, "ii= " & ii & " rowindex= " & RowIndex & " icol= " & icol
         xsheet.Cells(RowIndex, icol).Value = outarray(iarray)
         icol = icol + 1
      Next iarray
      RowIndex = RowIndex + 1

   Next ii



   Line Input #1, InLine
   Line Input #1, InLine  ' get the i indices
   'Print #10, InLine & "  getting indices"
   split_on_blanks InLine, outarray
   icol = StartColumn
   For i = 2 To UBound(outarray)


      xsheet.Cells(RowIndex, icol).Value = outarray(i)
      icol = icol + 1
   Next i
   RowIndex = RowIndex + 1
   icol = StartColumn
```

```
                Line Input #1, InLine  ' get coordinates
                split_on_blanks InLine, outarray

                For i = 2 To UBound(outarray)
                   xsheet.Cells(RowIndex, icol).Value = outarray(i)
                   icol = icol + 1
                Next i
                StartColumn = icol - 1
                Line Input #1, InLine
                Line Input #1, InLine

                iblock = iblock + 1
              End If  ' if for the nth block


              If iblock > 9 Then  ' get out of loop and start next file
                 Exit Do
              End If
            Loop ' inner loop
          End If

          If iblock > 9 Then ' get out of loop and start next file
             Exit Do
          End If

        Loop ' loop to read in file

        Close

    Next subf
End Sub
```

Private Sub summarytables()

```
Application.DisplayAlerts = False
Application.ScreenUpdating = False

Dim SheetTest
Set SheetTest = New RegExp
SheetTest.Pattern = "TI"

' set up the nodes to get the data from
Dim IGrout, JGrout, JDirt
IGrout = Array(37, 40, 43, 44, 45, 47, 48, 49, 51, 52, 53, 54, 56, 58, 60, 62, 64, 66, 69, 73)
'JGrout = Array(5, 10, 15, 20, 25, 30, 35, 40, 45, 50) ' this line is proflow node numbers, next line is sheet index
JGrout = Array(90, 85, 80, 75, 70, 65, 60, 55, 50, 45)

'JDirt = Array(72, 76, 80, 84) 'same deal as above
JDirt = Array(23, 19, 15, 11)

For Each sheetname In ActiveWorkbook.Sheets ' if the sheet already exists, delete it
   If sheetname.Name = "Summary Table Grout" Then
      sheetname.Delete
   End If
Next sheetname

Set xsheet = Sheets.Add
xsheet.Name = "Summary Table Grout"

'do column mheading
xsheet.Cells(1, 1).Value = "Time Period"
icol = 2
For i = 0 To UBound(IGrout)
```

```
   xsheet.Cells(1, icol).Value = "i=" & IGrout(i)
   icol = icol + 1
Next i

' loop through all hte sheets
irow = 2
For Each sheetname In ActiveWorkbook.Sheets
   If SheetTest.Test(sheetname.Name) Then  ' test to see if a time period data sheet
      icol = 2
      xsheet.Cells(irow, 1).Value = sheetname.Name

      For i = 0 To UBound(IGrout)
         'get an average value for the horizontal
         hsum = 0
         For j = 0 To UBound(JGrout)
            ' column index is offset by 2 because its data is node number
            hsum = hsum + sheetname.Cells(JGrout(j), IGrout(i) + 2).Value
         Next j
         havg = hsum / UBound(JGrout)
         xsheet.Cells(irow, icol).Value = havg
         icol = icol + 1
      Next i
      irow = irow + 1
   End If
Next sheetname

'*************************************************************************************
******
' do it for the dirt, difference is the j index
For Each sheetname In ActiveWorkbook.Sheets ' if the sheet already exists, delete it
   If sheetname.Name = "Summary Table Dirt" Then
      sheetname.Delete
```

```
    End If
Next sheetname

Set xsheet = Sheets.Add
xsheet.Name = "Summary Table Dirt"

'do column mheading
xsheet.Cells(1, 1).Value = "Time Period"
icol = 2
For i = 0 To UBound(IGrout)
    xsheet.Cells(1, icol).Value = "i=" & IGrout(i)
    icol = icol + 1
Next i

' loop through all the sheets
irow = 2
For Each sheetname In ActiveWorkbook.Sheets
    If SheetTest.Test(sheetname.Name) Then  ' test to see if a time period data sheet
        icol = 2
        xsheet.Cells(irow, 1).Value = sheetname.Name

        For i = 0 To UBound(IGrout)
            'get an average value for the horizontal
            hsum = 0
            For j = 0 To UBound(JDirt)
                ' column index is offset by 2 because its data is node number
                hsum = hsum + sheetname.Cells(JDirt(j), IGrout(i) + 2).Value
            Next j
            havg = hsum / UBound(JDirt)
            xsheet.Cells(irow, icol).Value = havg
            icol = icol + 1
        Next i
```

```
      irow = irow + 1
   End If
Next sheetname
'*************************************************************************************
********

' do it for the UZ

Dim IUZ, JUZ
IUZ = Array(3, 7, 11, 15)
'JUZ = Array(5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 90)  porflow nodes
JUZ = Array(90, 85, 80, 75, 70, 65, 60, 55, 50, 45, 40, 35, 30, 25, 20, 15)

For Each sheetname In ActiveWorkbook.Sheets ' if the sheet already exists, delete it
   If sheetname.Name = "Summary Table Unsat Zone" Then
      sheetname.Delete
   End If
Next sheetname

Set xsheet = Sheets.Add
xsheet.Name = "Summary Table Unsat Zone"

'do column mheading
xsheet.Cells(1, 1).Value = "Time Period"
icol = 2
For i = 0 To UBound(IUZ)
   xsheet.Cells(1, icol).Value = "i=" & IUZ(i)
   icol = icol + 1
Next i

' loop through all the sheets
irow = 2
```

```
For Each sheetname In ActiveWorkbook.Sheets
   If SheetTest.Test(sheetname.Name) Then  ' test to see if a time period data sheet
      icol = 2
      xsheet.Cells(irow, 1).Value = sheetname.Name

      For i = 0 To UBound(IUZ)
         'get an average value for the horizontal
         hsum = 0
         For j = 0 To UBound(JUZ)
            ' column index is offset by 2 because its data is node number
            hsum = hsum + sheetname.Cells(JUZ(j), IUZ(i) + 2).Value
         Next j
         havg = hsum / UBound(JUZ)
         xsheet.Cells(irow, icol).Value = havg
         icol = icol + 1
      Next i
      irow = irow + 1
   End If
Next sheetname
End Sub
```

Private Sub FlipGrout()

```
Application.ScreenUpdating = False
Set grout = Sheets("Summary Table Grout")
Set flipped = Sheets("Summary Grout Flipped")
For icol = 2 To 21
   iflip = 45
   For irow = 2 To 45
      flipped.Cells(iflip, icol).Value = Abs(grout.Cells(irow, icol).Value)
      iflip = iflip - 1
   Next irow
```

```
Next icol
'MsgBox "flipgrout"

End Sub
Private Sub FlipDirt()

Application.ScreenUpdating = False
Set dirt = Sheets("Summary Table Dirt")
Set flipped = Sheets("Summary Table Dirt Flipped")
For icol = 2 To 21
   iflip = 45
   For irow = 2 To 45
      flipped.Cells(iflip, icol).Value = Abs(dirt.Cells(irow, icol).Value)
      iflip = iflip - 1
   Next irow
Next icol
'MsgBox "flipdirt"
End Sub
Private Sub FlipUZ()

Application.ScreenUpdating = False
Set dirt = Sheets("Summary Table Unsat Zone")
Set flipped = Sheets("Summary Table Unsat Flipped")
For icol = 2 To 6
   iflip = 45
   For irow = 2 To 45
      flipped.Cells(iflip, icol).Value = Abs(dirt.Cells(irow, icol).Value)
      iflip = iflip - 1
   Next irow
Next icol
'MsgBox "flipuz"
End Sub
```

```
Private Sub getwall()
' use same value for all , so pick one from the middle, porflow(i,j) = (59,30), cell(30,61)
Application.ScreenUpdating = False
Set wf = Sheets("WallFlux")



For i = 1 To 44
   If i < 10 Then
      xx = "TI0" & i
      Set pf = Sheets(xx)
   Else
      xx = "TI" & i
      Set pf = Sheets(xx)
   End If

   wf.Cells(i, 1).Value = xx
   wf.Cells(i, 2).Value = pf.Cells(37, 58 + 2).Value * -1#

Next i
'MsgBox "done"
End Sub
```

Private Sub split_on_blanks(InLine, outarray)

```
ReDim outarray(1)
xcnt = 0
Set reg = New RegExp
reg.Global = True
reg.IgnoreCase = True
reg.Pattern = "\s*(\S+)"
Set splitx = reg.Execute(InLine)
```

```
For jmatch = 0 To splitx.Count - 1
   Set xblank = splitx(jmatch)
   For Each xword In xblank.SubMatches
       xcnt = xcnt + 1
       ReDim Preserve outarray(xcnt)
      outarray(xcnt - 1) = xword

   Next xword
Next jmatch

End Sub
```

Appendix B Plume Function Description

The report can be read in its entirety by double-clicking its first page. It resides in this report as an embedded pdf**.**

Notes on the GoldSim Plume Function

2 October 2008

Prepared by
John Tauxe
Neptune and Company, Inc.

1. Title: Notes on the GoldSim Plume Function			
2. Filename: GoldSim Plume function math.doc			
3. Description: This documents calculation details of the GoldSim plume function, as interpreted from the source code: PlumeFuncNode.cpp			
	Printed Name	Signature	Date
4. Originator	John Tauxe		16 Sep 2008
5. Reviewer	Ian Miller		29 Sep 2008
6. Remarks: 24 Sep 08 – Corrected a typo in the in-line equations in Section 2.0, and added the image source illustration in Figure 2. – JT 28 Sep 08 – Added Figure 1, and made clarifying edits to text. – IM			

CONTENTS

FIGURES

1.0 Introduction

The plume() function in GoldSim produces a value between 0 and 1, which is to be used as a multiplier applied to the concentration at the end of a Pipe Pathway element, in order to account for lateral and vertical dispersion when the Pipe is intended to represent an aquifer, for example. These are not allowed in the Pipe calculation, which has a specified cross-sectional area and circumference around impermeable walls—more like a straw than an aquifer. A point of observation is identified in three dimensions, X_L, X_T, and X_V, corresponding to longitudinal, transverse (lateral) and vertical locations in a coordinate system based on the center of the upstream edge of the source zone and aligned with the flow of water in the Pipe. The longitudinal location X_L corresponds to the end of the Pipe. Figure indicates the geometry that is assumed.

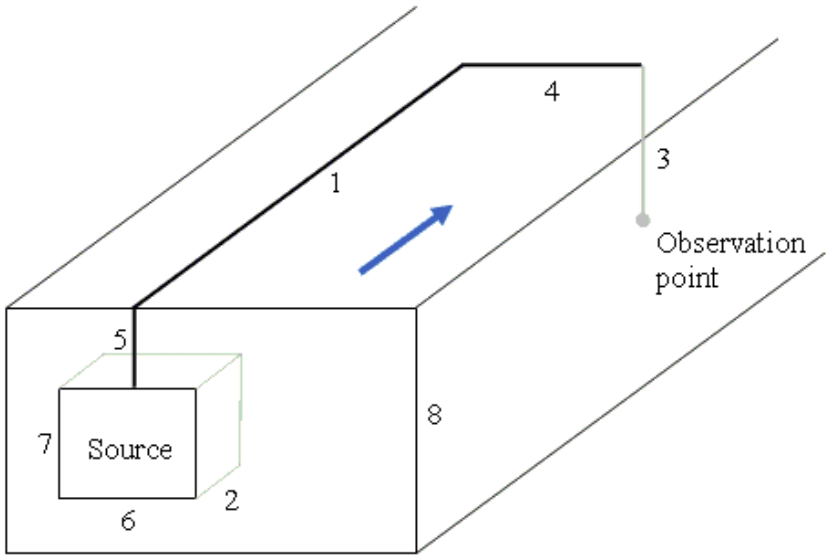

Figure 1. Plume function geometry (courtesy of GoldSim Technology Group).

It should be noted that as of GoldSim v9.70, the plume() function was upgraded to handle inputs that are GoldSim vectors and matrices. These calculations are simply term-by-term calculations, as if the plume() function had been applied to each term independently. This is useful in GoldSim programming, since a single use of the function can be used to determine plume corrections at a large number of independent locations.

GoldSim contaminant transport (CT) models developed in support of Performance Assessments (PAs) at the Savannah River Site (SRS) have applied this plume "correction" to concentrations at the end of a "column" of Cell Pathway elements, also used to represent an aquifer. This technique offers several advantages to the Pipe, the discussion of which is beyond the scope of this document. For simplicity, references to "Pipe" in the remainder of this document could be replaced with "column of Cells", except where noted.

The plume() function has eleven arguments, in the following order. The corresponding items in Figure 1 are shown in brackets:

- X_L length of the Pipe, or, in a Cell column, the longitudinal position of the observation point, which should be made coincident with the center of the Cell nearest the desired column length [1]
- A cross-sectional area of the Pipe
- L_s length of the source parallel to the flow direction [2]
- X_V vertical position of the observation point [3]
- X_T transverse position of the observation point [4]
- D_s vertical depth to the top of the source from the top of the aquifer [5]
- W_s width of the source, transverse to the aquifer flow [6]
- b_s thickness of the source [7]
- b thickness of the aquifer [8]
- α_T dispersivity in the transverse direction
- α_V dispersivity in the vertical direction

All these inputs except the area have dimensions of L (length), with the area A in L^2. GoldSim evaluates all arguments to ensure consistent dimensions and units, so this need not be addressed further in this discussion.

The dispersivity in the longitudinal direction does not enter this equation directly, since it is included in the Pipe solution of concentration. For the column of Cells, longitudinal dispersion is handled numerically. Other time- and chemical-related parameters are also handled in the contaminant transport (CT) calculations, as discussed in the following section.

2.0 Assumptions and Boundary Conditions

A clever aspect of the plume() function is that it does not include a full-blown plume calculation, which would necessarily involve transport-relevant parameters like retardation, decay and ingrowth, and such. These are all handled by the transport Pathways (Pipe or Cell column), so all the plume() function need do is modify the spatial distribution of contamination. It does so by providing the scalar by which the end-of-the-Pipe concentration is modified, effectively dispersing the straw-like Pipe calculation in space.

Boundary conditions include the assumption of zero concentration at infinite distance, and a special handling of the output of the GoldSim Pipe element. The Pipe element provides a contaminant concentration (in dimensions of M/L^3) at the end of the Pipe expressed as a mass flux discharge per flow rate, that is, $\frac{\text{mass flux discharge (M/T)}}{\text{Pipe flow rate (L}^3\text{/T)}}$. As discussed below (in Section 3.1.1) the factor calculated by the plume function includes the cross-sectional area of the Pipe, A, so that the calculated Pipe concentration is thereby a mass per volume (M/L^3):
$\frac{\text{areal mass flux (M/TL}^2\text{)}}{\text{Darcy velocity (L/T)}}$.

3.0 Functionality of PlumeFuncNode.cpp

PlumeFuncNode.cpp is the C++ source code for the plume() function, kindly provided by GoldSim Technology Group for the purposes of this documentation.

3.1 Setting Up

After running all arguments through GoldSim Eval() function, which assures that they all have proper numeric and dimensionally consistent values, and after determining whether the arguments are scalars, or vectors or matrices of scalar values, the plume() function begins constructing the answer that it will return – the value (or vector or matrix of such values) between 0 and 1 that is to be multiplied by the end-of-the-Pipe concentration to arrive at a concentration at the designated observation location(s). This result is called dilute within the function.

3.1.1 Initial Value

The value of dilute is calculated in parts, with all parts multiplied in the end, as described in Section 3.4. Initially, we set $dilute_{init}$ to the cross-sectional area of the Pipe, A. As the value of dilute is modified for dispersion effects in the subsequent calculations, it is twice divided by a length, leaving it dimensionless. In effect, this counteracts the use of cross-sectional area in the Pipe calculation. In the development of this calculation, we shall keep track of the various parts of the equation for dilute that are proffered. So far, we have

$$dilute_{init} = A, \tag{1}$$

where A is the cross-sectional area of the Pipe or column of Cell Pathway elements. Equation (1) converts the end-of-the-Pipe concentration to the total mass flux leaving the Pipe divided by the total flow in the pipe, as shown in Section 2.0. In order to convert that to a concentration at an observation point, it needs to be converted to the mass flux rate across a (small) unit area at the monitored location, divided by the Darcy velocity. This is done by multiplying by the 2-D spatial density at the observation point, for the density function in the plane perpendicular to the flow, of a unit mass input uniformly distributed over the source region, as dispersed after traveling the specified advective transport distance.

3.1.2 Relative Length Adjustment

The length X_L of the Pipe is assumed to be measured from the upstream end of the source. It is then modified in order to center the source at the longitudinal "origin". The following cases are considered:

Case: if $X_L <= 0$, then the point of observation is upstream of the source, and

$$dilute_{init} = 0 \tag{2}$$

At this point, X_L is also set to $X_L = 1$ (with units of m implied), to preclude divide-by-zero errors.

Case: if $X_L > L_s$, then the point of observation is downstream of the source, and X_L is adjusted thus

$$X_L^* = X_L - \frac{L_s}{2} \tag{3}$$

so that the "source" end of the Pipe is moved to the center of the source. This adjusted length is denoted X_L^*.

Case: if neither of the above cases is true, then $X_L <= L_s$, meaning that the Pipe is shorter than the source. In this case, X_L is halved:

$$X_L^* = \frac{X_L}{2} \tag{4}$$

This, in effect, truncates the source to be the length of the Pipe, and the upstream end is moved to the center of the source. In the latter two cases, the value of $\text{dilute}_{\text{init}}$ remains set equal to A.

3.2 Transverse Dispersion

In accounting for transverse dispersion, various cases are considered: a narrow (point) source, a wide source, and the case where the observation point is too far afield to be considered part of the plume. The cases are presented as follows:

3.2.1 Narrow (Point) Source

A narrow source triggers the point-source solution for transverse dispersion. In comparing the source width W_s to the transverse standard deviation of the plume $\sqrt{2\alpha_T X_L^*}$, a narrow source solution is used if

$$W_s < \frac{\sqrt{2\alpha_T X_L^*}}{100}, \tag{5}$$

where the value of 100 is "hard coded" into the function, based on the judgment of the developer.

In this case, the point source solution is used, and the transverse portion of dilute, hereafter called dilute_T, assumes the form of the normal distribution density function:

$$dilute_T = \frac{\exp\left[-\frac{\left(\frac{X_T}{\sqrt{2\alpha_T X_L^*}}\right)^2}{2}\right]}{\sqrt{2\pi}\sqrt{2\alpha_T X_L^*}} = \frac{\exp\left(-\frac{X_T^{\,2}}{4\alpha_T X_L^*}\right)}{\sqrt{4\pi\alpha_T X_L^*}} \tag{6}$$

Keep in mind that $dilute_T$ is to be multiplied by $dilute_{init}$ as defined in Section 3.1 in order to arrive at the final answer.

3.2.2 Wide (General) Source

A "wide" source, the general full solution, is indicated if the following is true:

$$W_s < 1{,}000{,}000 \times \sqrt{2\alpha_T X_L^*} \tag{7}$$

This applies in the case where a source is not negligibly wide, and dispersion cannot be ignored. In this case, the point source solution is integrated over the width of the source, and $dilute_T$ is

$$dilute_T = \frac{\operatorname{erf}\left(\frac{X_T + \frac{W_s}{2}}{\sqrt{4\alpha_T X_L^*}}\right) - \operatorname{erf}\left(\frac{X_T - \frac{W_s}{2}}{\sqrt{4\alpha_T X_L^*}}\right)}{2W_s} \tag{8}$$

3.2.3 Negligible Transverse Dispersion

If neither of the previous two situations occur then the transverse dispersion is negligible (i.e. the plume does not spread at all), and the observation point is tested to see if it inside or outside of the plume. The test for being outside the plume is if the transverse location of the observation point is greater than half the source width from the plume centerline:

$$|X_T| > \frac{W_s}{2} \tag{9}$$

In effect, if the observation location is further from the centerline than the edge of the source, then it is out of the plume. If equation (9) holds true, then the point is outside the plume, and $dilute_T$ (and therefore dilute, as described in Section 3.4) is set to zero. Otherwise $dilute_T$ is set equal to the source width, W_s.

3.3 Vertical Dispersion

In the case of dispersion in the vertical direction, again perpendicular to flow, five cases are considered: a fully-mixed aquifer, a thin source, a thick source, no source geometry adjustment, and the case where the observation point is too shallow or deep to be considered part of the plume. Since the flow field is assumed to have natural boundaries on the top and bottom, but not on the sides, the calculation of concentrations at vertical locations within the aquifer takes advantage of superposed image source mathematics. Note that unlike the transverse case, where the source is always centered horizontally, the vertical position of the source may be anywhere within the flow field. Similar to the transverse case, however, the vertical aspect of the source may be relatively thin or thick compared to vertical dispersion. The cases are presented as follows:

3.3.1 Fully-Mixed Aquifer

If an aquifer (or, more generically, the flow field) is sufficiently thin in comparison to dispersion in the vertical direction, then the flow field can be assumed to be fully mixed in the vertical direction, and no image sources are used. That is, if we compare the aquifer thickness b to the vertical standard deviation of the plume $\sqrt{2\alpha_V X_L^*}$, and

$$b < \frac{\sqrt{2\alpha_V X_L^*}}{3} \tag{10}$$

(where again the value of 3 is chosen by the developer) then dilute is merely divided by that thickness, so that the vertical term $dilute_V$ becomes

$$dilute_V = \frac{1}{b} \tag{11}$$

3.3.2 Use of Image Sources

If the aquifer is not sufficiently thin to be considered fully-mixed, it is examined for suitability in using nine image sources to refine the vertical distribution of contaminants, as illustrated in Figure 2. In effect, the plume from each image source, identical to the original source, is offset and its contribution to the concentrations in the aquifer are superimposed. This has the same effect as "bouncing" the original plume off the top and bottom of the aquifer.

Image sources are invoked if, analogous to equation (7),

$$b < 1{,}000{,}000 \times \sqrt{2\alpha_V X_L^*} \tag{12}$$

where the value of 1,000,000 is selected by the developer. If this inequality is not true, then no image sources are used, and no further changes are made to dilute for vertical dispersion (and the following sections testing for a thin or thick source are also precluded). Like the transverse case presented in equation (7), equation (12) is true only for very small dispersivities.

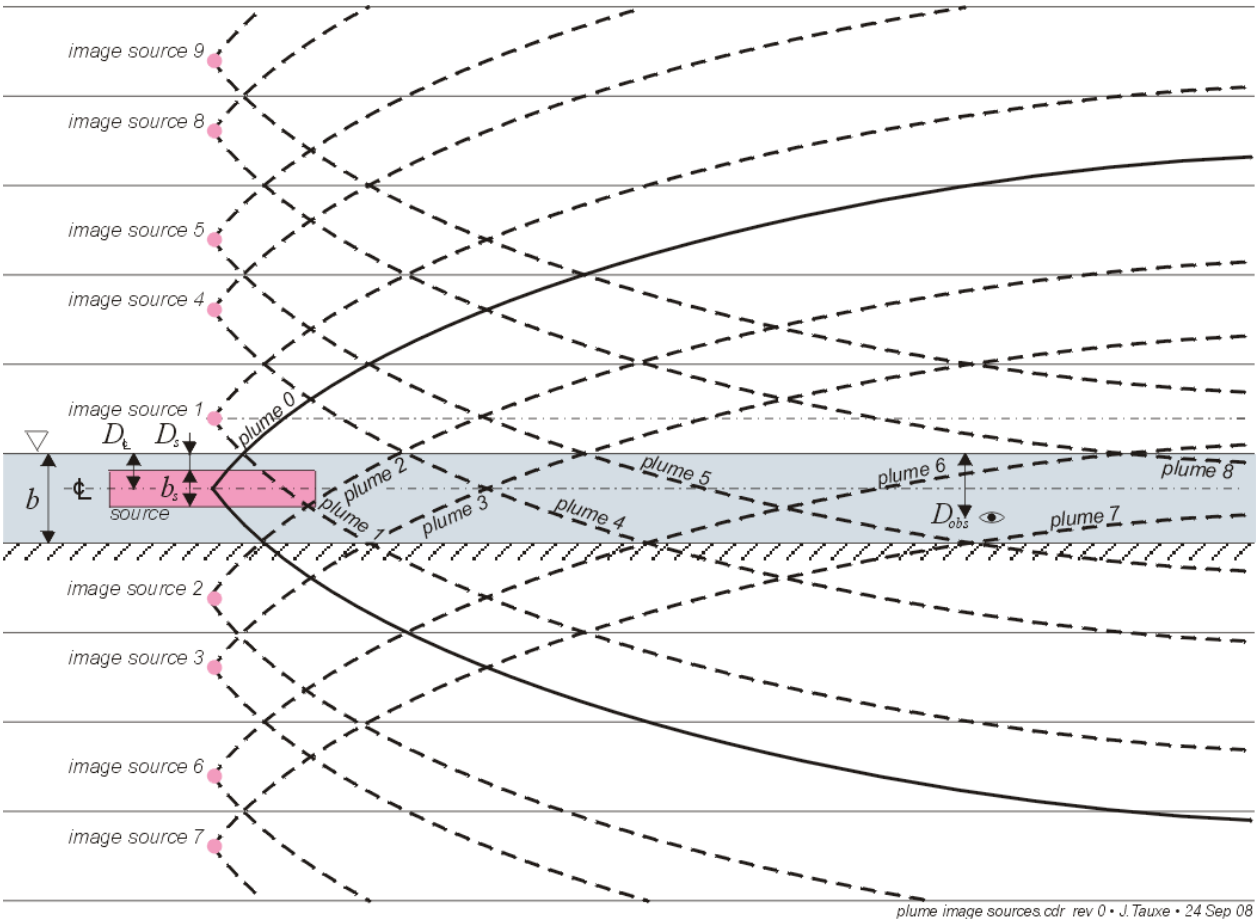

Figure 2. Placement of image sources relative to the aquifer and actual source.

3.3.2.1 Thin Source

A thin source triggers the point-source solution for vertical dispersion, expanded to include the additional image sources (the development of which is not covered here). In comparing the aquifer thickness b to the vertical standard deviation of the plume $\sqrt{2\alpha_V X_L^*}$, a thin source solution is used if

$$b < \frac{\sqrt{2\alpha_V X_L^*}}{100} \tag{13}$$

In this case, the point source solution is used with i images at vertical locations X_{Vi} relative to the source's vertical centerline, and dilute_V, is the summation of a number of normal distribution density terms:

$$dilute_V = \frac{\sum_i \exp\left(-\frac{X_{V_i}^{\ 2}}{4\alpha_V X_L^*}\right)}{\sqrt{4\pi\alpha_V X_L^*}} \tag{14}$$

Here dilute_V is to be multiplied by dilute_{init} as defined in Section 3.1 and dilute_T from Section 3.2 in order to arrive at the final answer.

If the criterion for using images is met, that is, equation (12) is true, and that for a thin source is not met, that is, equation (13) is false, then the source is assumed to be relatively thick.

3.3.2.2 Thick Source

In the case of a source that is relatively thick in comparison to vertical dispersion, the point source solution is integrated over the thickness of the source b_s and is summed over i image sources, so that dilute_V is

$$dilute_V = \frac{\sum_i \left[\text{erf}\left(\frac{X_{V_i} + \frac{b_s}{2}}{\sqrt{4\alpha_V X_L^*}}\right) - \text{erf}\left(\frac{X_{V_i} - \frac{b_s}{2}}{\sqrt{4\alpha_V X_L^*}}\right)\right]}{2\,b_s} \tag{15}$$

3.3.3 Negligible Vertical Dispersion

In the case where we still do not assume vertical mixing and also do not invoke the use of image sources, that is equations (10) and (12) are false, then a final test is made to determine whether the observation point is vertically outside the plume. If the vertical location of the observation point relative to the vertical centerline of the source is greater than half the source thickness, expressed as

$$\left|X_V\right| > \frac{b_s}{2} \tag{16}$$

then the observation point is considered to be out of the plume and dilute_V (and therefore dilute, as described in Section 3.4) is set to zero. Otherwise dilute_V is set equal to the source thickness.

3.4 Final Assembly

Having completed the various multiplicative factors involved in the calculation of dilute, we assemble the final calculation as the product

$$dilute = dilute_{init} \times dilute_T \times dilute_V \,. \tag{17}$$

In some cases, one or more of these terms may be equal to unity, in effect removing its influence, or zero, in effect making the entire calculation equal to zero.

The entire process is repeated for each item in the vector or matrix, if the plume() function is fed them rather than a collection of scalar arguments.

4.0 Implications for Use of the plume() Function at SRS

At the SRS, GoldSim PA models use the plume() function in the calculation of saturated zone contaminant transport from beneath the various sources (trenches in E-Area, high-level waste tanks in F- and H-Areas, and Saltstone vaults in Z-Area) to a number of potential locations for receptors' groundwater wells, generally located 100 m from the collection of sources. Each plume is modeled with a single horizontal "column" of Cell Pathway elements, and the plume() function is used to derive a PlumeCorrectionFactor for each source. Each well location is transposed into coordinates relative to each source, and the plume() function is applied to that longitudinal and transverse X_L and X_T. The aquifer is assumed to be fully mixed vertically, resulting from the assumption of a fully-penetrating water well. The concentration at each well is the superposition of the contributions from each source. Since each source originates in the unsaturated zone above the aquifer, contamination is introduced at the top of the aquifer, using the same areal footprint as the source (i.e. trench, tank, or vault).

So, the assumptions in invoking the plume() function in the SRS PA models are that there is no explicit vertical dispersion (since complete vertical mixing is assumed), and that horizontal transverse dispersion is controlled only by the supplied dispersivity α_T. Parameters supplied to the plume() function are therefore

- X_L length of the Cell column from the upstream end of the source to the receptor's well
- A cross-sectional area of the Cell column (source width × Cell column thickness)
- L_s length of the source parallel to the flow direction
- X_V vertical position of the observation point (set to 0 m)
- X_T transverse position of the observation point, perpendicular to the flow centerline for a given source, at X_L
- D_s vertical depth to the top of the source from the top of the aquifer (set to 0 m)
- W_s width of the source, transverse to the aquifer flow
- b_s thickness of the source (set to 0 m)
- b thickness of the aquifer
- α_T dispersivity in the transverse direction
- α_V dispersivity in the vertical direction (set to an arbitrarily large value)

For this specific set of input arguments, the plume() function will perform the following calculation:

First, $\text{dilute}_{\text{init}}$ is set to the cross-sectional area in equation (1):

$$dilute_{init} = A. \tag{18}$$

In all cases, the receptor wells are at least 100 m from the downstream edge of the source, so $X_L > L_s$ and equation (3) adjusts the assumed length of the column to be

$$X_L^* = X_L - \frac{L_s}{2}. \tag{19}$$

The value of L_s varies with the disposal facility, and can be as large as 200 m, reducing the column length X_L from, for an extreme example, 300 m to X_L^* = (300 – 200/2) m = 200 m, a reduction of 33%. At the other extreme, the tanks are circular with a diameter of about 23 m, and the distance to the well may be as large as 310 m, so the column length X_L would be reduced from 333 m to X_L^* = (333 – 23/2) m = 312.5 m, a reduction of only about 6%.

In considering transverse dispersion, we consider the width of the source (again ranging from about 23 m to 200 m, and the transverse dispersivity. Currently, the transverse dispersivity α_T is expressed as about 1/3 of the longitudinal dispersivity α_L, which in turn is a ratio of the column length, about X_L / 27.5, or $\alpha_T \approx X_L$ / 82.5.

To test the possibility of a narrow source, examine the narrowest source and the longest path length, or 23 m and 310 m, respectively. In this case, plume() would be supplied the arguments X_L = 310 m, W_s = 23 m, and $\alpha_T \approx$ 3.8 m.

Since the value for X_L is adjusted internally by plume() (in this case to about 300 m), the standard deviation of the transverse dispersion in the plume is about $\sqrt{2\alpha_T X_L^*} = \sqrt{2 \cdot 3.8\,\text{m} \cdot 300\,\text{m}} = 48\,\text{m}$. The test for a narrow source comes from equation (5):

$$W_s < \frac{\sqrt{2\alpha_T X_L^*}}{100} \quad \text{or} \quad 23\,\text{m} \overset{?}{<} \frac{48\,\text{m}}{100} \tag{20}$$

Clearly, 23 m > 0.48 m, so the narrow source test fails. Even when the SRS models are modified to accommodate contaminant transport to groundwater seeps a mile or more away, the narrow source test will fail. We may also reconsider the estimation of dispersivities, basing them on groundwater model calibrations rather than rough literature reviews. For now, however, suffice it to say that the sources are not classified as narrow.

Neither would they be expected to be particularly wide, though the "widest" aspect ratio of source to receptor locations could be at the Saltstone facility, where the Type I and IV vaults are 200 m "wide", and may be as close as 100 m from the receptors' well. In this case, $\sqrt{2\alpha_T X_L^*} = \sqrt{2 \cdot 1.2\,\text{m} \cdot 100\,\text{m}} = 16\,\text{m}$.

However, if we consider the definition of a "wide" source using equation (7), we see that

$$W_s < 1{,}000{,}000 \times \sqrt{2\alpha_T X_L^*} \quad \textit{or} \quad 200\,\text{m} \overset{?}{<} 16 \times 10^6\,\text{m}. \tag{21}$$

Certainly, this is true, and would be for all cases. It seems that for the purposes of calculations, the plume() function considers all the sources as wide sources. If that is the case, then equation (8) would be used to calculate the transverse dispersion, dilute_T.

The vertical dispersivity α_V in the SRS models is supplied to plume() as an arbitrarily large number. Therefore the standard deviation of the vertical dispersivity, $\sqrt{2\alpha_V X_L^*}$ is also very large. The test for a fully-mixed aquifer in equation (10) therefore passes, and equation (11) is invoked: $dilute_V = \frac{1}{b}$.

In the SRS application, therefore, as long as the arguments X_V, D_s, and b_s, are all 0 m and the vertical dispersivity α_V is very large, the final expression for dilute (used as the PlumeCorrectionFactor in the SRS models) is based on equation (8), including the column cross-sectional area A and dividing by the aquifer thickness b, and with the substitution of the modified $X_L - \frac{L_s}{2}$ for column length X_L, just to keep the equation in terms of the original inputs

$$A \times \frac{\operatorname{erf}\left[\frac{X_T + \frac{W_s}{2}}{\sqrt{4\alpha_T\left(X_L - \frac{L_s}{2}\right)}}\right] - \operatorname{erf}\left[\frac{X_T - \frac{W_s}{2}}{\sqrt{4\alpha_T\left(X_L - \frac{L_s}{2}\right)}}\right]}{2bW_s} \tag{22}$$

Since through careful definition of the Cell column, the area A normal to flow equals the source width W_s times the aquifer thickness b, the equation for PlumeCorrectionFactor simplifies to

$$\frac{\operatorname{erf}\left[\frac{X_T + \frac{W_s}{2}}{\sqrt{4\alpha_T\left(X_L - \frac{L_s}{2}\right)}}\right] - \operatorname{erf}\left[\frac{X_T - \frac{W_s}{2}}{\sqrt{4\alpha_T\left(X_L - \frac{L_s}{2}\right)}}\right]}{2}. \tag{23}$$