# OVERVIEW AND SCOPE

| Acronym: | MODPATH | HISI ID: | [] | Software Grade: | [C] |
| --- | --- | --- | --- | --- | --- |

This Software Test Plan (STP) is required by PRC-PRO-IRM-309, *Controlled Software Management*, to define the testing requirements for MODPATH and related codes for CH2M HILL Plateau Remediation Company (CHPRC) intended use in risk and model integration work.

MODPATH and related codes are managed as acquired, commercial off-the-shelf (COTS) software applications. The CHPRC plans to use this for calculations of short- and long-term subsurface particle tracking in the unconfined aquifer at the Hanford Site at several scales. The software will run on desktop computers, scientific workstations, and large computer clusters.

The test procedures apply to all of the software being tested, although tests are limited to representative problems for the class of problems expected to be solved at the Hanford Site to demonstrate applicability for intended use by the CHPRC Integration and Assessments Group.

The MODFLOW calculational software configuration items tested in this STP are:

* **MODPATH**
* **MODPATH3DU** (variant of MODPATH which calculates particle pathlines on unstructured flow grids)

Supporting software, as identified in [CHPRC-00257, *MODFLOW and Related Codes Functional Requirements Document* (Nichols 2009b) (FRD)], are not tested in this STP because these were not classified as safety software or graded to a level requiring testing.

[Controlled use of this software is managed by CHPRC-00258, *MODFLOW and Related Codes Software Management Plan* (Nichols 2009c) (SMP). The major milestone of the test procedures is issuance of CHPRC-00261, *MODFLOW and Related Codes Acceptance Test Report* (Nichols 2009a) (ATR) that will enable authorization for use for the above software to be granted.]

# TEST REQUIREMENTS

## Approach

The general approach for this STP is to test the MODPATH and MODPATH3DU codes each against test problems specifically chosen with the following attributes: 1) availability of an analytical solution or [comparison to someone else’s solution] to facilitate comparison to a known solution, 2) sufficiently difficult for numerical approximation techniques to resolve that the test will provide not only evidence of a correct solution but a measure of the degree of error in the solution, and 3) ability to substitute Hanford-specific parameter values where possible to facilitative demonstration of applicability for intended use. As such, acceptance criteria necessarily will rely on professional judgment at least in part to evaluate if the solution obtained from the software that implements approximate numerical solution techniques is robust enough compared to the analytical solution or the case replicated to merit acceptance. It is possible to pose easier problems for which the numerical codes will produce exactly the correct solutions, but validation against a difficult-to-resolve problem is more informative. Acceptance criteria are deliberately chosen to avoid problems that arise from comparison of extremely small values that can give large percentage differences but don’t represent any significant difference.

The sequence of testing activities is discussed in the presentation of the test cases in Section 3.3.

## Deliverables

This STP specifies test design, test cases, and test procedures for acceptance and installation testing of MODPATH and related codes.

For acceptance testing, test results will be documented in an ATR using the format provided in the ATR format guidance referenced in PRC-PRO-IRM-309.

For installation testing, results will be documented in Software *Installation & Checkout Form* (A-6005-149).

## Control Procedures

The test cases in this STP are relatively simple and efficient to conduct and do not require formal tracking or control procedures. The test case logs will be used to document all steps and track results.

## Acceptance Criteria

Acceptance criteria will specify objective numerical tolerances and subjective professional judgment metrics for specific tests that must be met by the code in solving classical groundwater problems with Hanford-specific properties in order to pass. These criteria are specified with the test description in Section 3.3.

## Testing Tasks

[The tasks necessary to prepare for and perform testing include code installation (which may require compiling and linking source code using a Fortran compiler, depending on test environment), file transfers, checking input/output files using a text editor program, copying model results from text files using a text editor, and pasting results into Excel spreadsheets used to solve the analytic solutions that provide the basis for validation and compare results.]

### Verification & Validation tasks

Tasks necessary to prepare for verification and validation testing are routine with respect to computer usage. MODPATH and related codes must be installed. Users will run a virus scan of the host computer at completion of the software installation, consistent with requirements of PRC-PRO-IRM-309.

For acceptance testing, the following tasks will occur:

1. Log – the results of test execution, the incidents observed, and any other events pertinent to the test are to be logged on the forms provided as attachments to this STP.
2. Setup – verify that the host computer uses a suitable operating system for the MODPATH and related software, and that these software are installed, registered as appropriate, and virus-checked. If this software is not installed, complete the installation per directions received from the appropriate provider and virus-check the host computer immediately following software installation.
3. Execution – run the acceptance tests.
4. Evaluation – compare results of the acceptance test cases to the baseline results and evaluate if acceptance criteria are met.
5. Contingencies – in the event that output obtained from the acceptance tests appear flawed, the user may identify and correct errors and restart the test. All such attempts and corrective actions taken will be logged. Return to step 3.
6. Documentation – complete the required documentation of testing in the ATR.
7. Preservation – archive the MODPATH and related codes software in [MKS Integrity]™[[1]](#footnote-1) to preserve the baseline as appropriate, along with the model file used for the acceptance test.
8. Review – independent technical review of test results is required and will be documented in the ATR.

For installation testing, the following tasks will occur:

1. Setup – verify that the host computer uses a suitable operating system for the MODPATH and related codes software, and that software is installed, registered as appropriate, and virus-checked. If the software is not installed, complete the installation and virus-check the host computer immediately following installation.
2. Execution – run the installation tests
3. Evaluation – Compare results of the installation test case to the baseline results and evaluate if acceptance criteria are met.
4. Documentation – complete the required documentation of installation testing.

If software problem(s) (that is, problems with the software itself, not input or other use errors) are identified during testing, a *Problem Report/Change Request* form (A-6005-146) will be prepared by the software tester in accordance with Sections 3.4 of PRC-PRO-IRM-309; the form will document as applicable the following information:

* Nature of required change (Minor/Major)
* Disposition and Proposed Corrective Action
* Impact Analysis
* Affected Hardware components
* Affected Baseline Documentation
* Affected software (e.g., libraries, databases)
* Testing Requirements

Submit the completed *Problem Report/Change Request* form to the software owner who will respond following the process identified in the SMP.

### Responsibilities

The software owner will:

* Conduct acceptance testing, or delegate this task to a qualified MODPATH software user
* Assign an independent technical reviewer who is not the person conducting the test to review and approve acceptance test results (the software owner may serve as the independent technical reviewer unless the software owner is also the software user).
* Ensure that acceptance testing results are documented in the ATR, obtain approvals, and issue the ATR.
* Respond to submission of *Problem Report/Change Request* forms as specified in the SMP.
* Maintain HISI entries related to software status, testing, and approval for use of MODFLOW and related codes as identified in PRC-PRO-IRM-309
* Archive a copy of the software and any vendor provided documents in the MKS Integrity™ configuration management system.

The independent technical reviewer will:

* Review this STP and ATR or installation test results for completeness, consistency, clarity and correctness.
* Confirm acceptable test results for acceptance testing.

The software user will do the following before using a software installation to produce results that will be reported in released documents or used for decision making purposes:

* Only use versions of MODPATH and related codes that are approved for use (following completion of acceptance testing).
* When installing the MODPATH and related codes, complete the installation test identified in this STP and documents the results using *Software Installation & Checkout Form* (A-6005-149).
* Ensuring that the installation test is repeated and results documented following software or hardware configuration changes to the host computer (e.g., operating system patches or upgrades)

### Risks and Contingencies

No major risks are identified with respect to this test plan. The MODPATH and related software has a long history of successful application to similar problems as CHPRC intends to use this software to solve and is the most widely used software in the world for this purpose.

# TEST CASES

## Identification

Test cases are sequentially numbered as Test Case 1, Test Case 2, etc.

## Pass/Fail Criteria

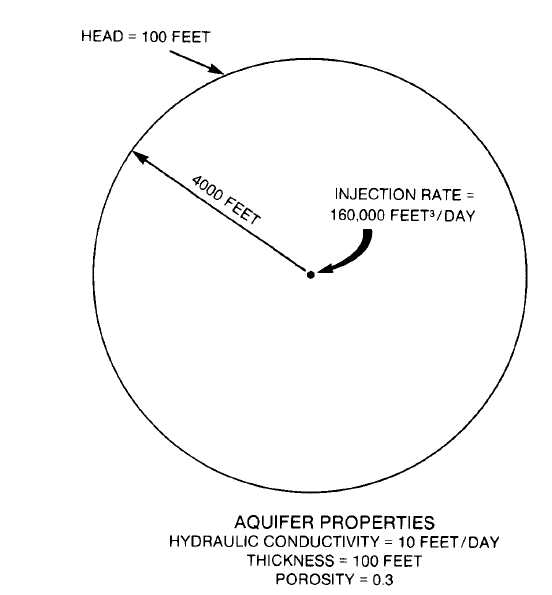
Specific pass/fail criteria for acceptance test cases are identified below in Section 3.3 for each case, and are set to ensure accurate solutions by the numerical solver in the tested software within an objective numerical tolerance. When the software is shown through the testing to meet these criteria, it will be considered to have passed the test.

## Test Cases

Four acceptance test cases and four installation test cases are detailed below with acceptance criteria and expected results for each. [The tests are applicable to both MODPATH and MODPATH3DU]; testing for each will be conducted and documented separately. Each of these test cases calculated the particle tracks over a flow field calculated using [either] MODFLOW [or MODFLOW3DU]. The python scripts create the input files for each of these tests [including for 3du?], executes the model runs, and post-processes the results utilizing the [plug-ins] FloPy, matplotlib, numpy, [and anything else?]. All [five] of these tests are executed via a batch file [name], which then outputs the results of the tests to “pass/fail.csv”, a document which contains the pass/fail status of each test.

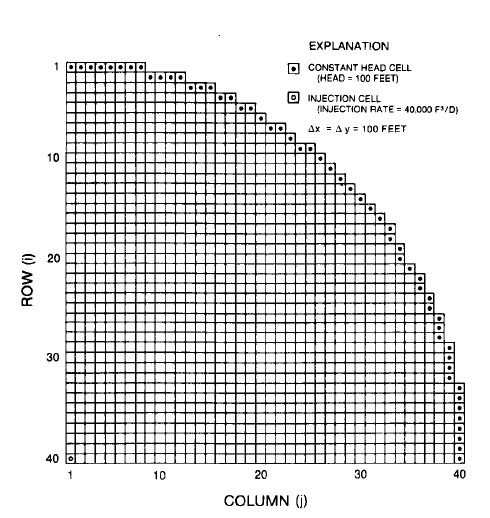
***Test Case 1: Forward Particle Tracking from an Injection Well***

The first test concerns forward particle tracking in a steady-state system with an injection well. This test is repeated from and compared against a test in Pollock, 1988, *Semianalytical Computation of Path Lines for Finite-Difference Models*. The injection well pumps at a constant rate of 160,00 ft3/day into a confined aquifer with a thickness of 100 ft, hydraulic conductivity of 10 ft/day (Figure 2‑1). The hydraulic head at a radial distance of 4,000 ft from the well is held constant at 100 ft (Figure 2‑2). The symmetry of the problem allows only one-fourth of the circular flow field be considered. Figure 2‑2 shows the finite-difference grid and boundary conditions used to approximate flow through one-fourth of the system shown in Figure 2‑1.



Source: Pollock, 1988

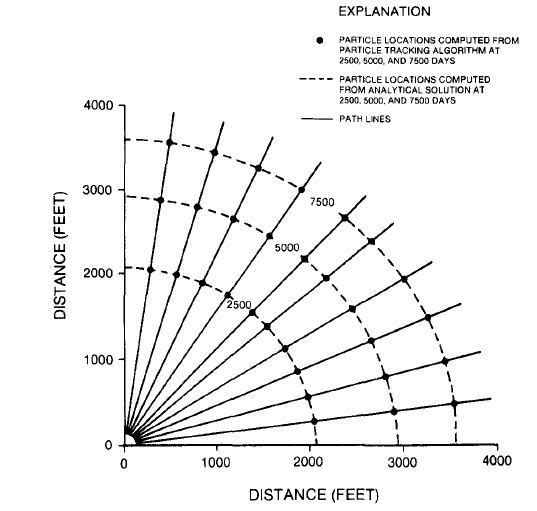
Figure 2‑1. Test Case 1 Layout



Source: Pollock, 1988

Figure 2‑2. Test Case 1 MODFLOW Setup

Ten particles were placed at a radial distance of 150 ft from the center of the well. Figure 2‑3 shows the positions of the particles after 2,500, 5,000, and 7,500 days, as calculated in Pollock, 1988. This figure was digitized and used as a guide for checking the model.



Source: Pollock, 1988

Figure 2‑3. Particle Location Over Time in Test Case 1

Model inputs are seen in Table 2‑1. Constant head cell locations are seen in Table 2‑2. The model has a single layer and has 40 rows and 40 columns each with a length and width of 100 ft. The thickness of the model is set to a uniform 100 ft. Constant head cells were assigned in the pattern shown in Figure 2‑2, and have a constant head of 100 ft. An injection well is located in the bottom left corner (row 40, column 1) and pumps at a rate of 40,000 ft3/day, representing a quarter of the pumping rate of the full-size model. The center of the well is located in the center of the cell. Starting head locations were placed in a 150-foot radius around the center of the cell in the lower left corner.

The python script which made the MODFLOW and MODPATH inputs and executed the MODFLOW and MODPATH simulations is “S01\_build\_pollock\_88.py”. Constant head cell locations are listed in “chb\_t1.csv”. Starting particle locations were calculated in “Write\_starting\_locations.py”. The model files are stored in /workspace, and post-processed model results are stored in /output. Within /output, /figures contains outputs of figures of the model results at 2,500, 5,000, and 7,500 days. The file “tc1\_results.csv” contains the information used to determine the pass or fail status of the test.

| Table 2‑1. Model Parameter Values for Test Case 1 | | | |
| --- | --- | --- | --- |
| Variable name | Variable Value | Units | Source |
| Number of rows | 40 | N/A | Pollock, 1988, Figure 6 |
| Number of columns | 40 | N/A | Pollock, 1988, Figure 6 |
| Height of rows | 100 | ft | Pollock, 1988, number of rows divided by radial distance of 4,000 ft |
| Width of columns | 100 | ft | Pollock, 1988, number of rows divided by radial distance of 4,000 ft |
| Number of layers | 1 | N/A | Pollock, 1988 |
| Thickness of layer | 100 | ft | Pollock, 1988 |
| Injection rate | 40,000 | ft3/day | Pollock, 1988 |
| Hydraulic conductivity (hk and vka) | 10 | ft/day | Pollock, 1988 for hk, vka assumed |
| Porosity | 0.3 | N/A | Pollock, 1988 |
| Constant head | 100 | ft | Pollock, 1988. Shape of the CHB was copied from Figure 6 |
| Starting head | 100 | ft | Assumed |
| Number of particles | 10 | N/A | Pollock, 1988 |
| Particle radial distance from center of the injection well | 150 | ft | Pollock, 1988 |
| Number of stress periods | 15 | N/A | Assumed |
| Stress period length | 500 | days | Assumed |
|  | | | |

| Table 2‑2. Constant Head Boundary Cell Locations | | | | |
| --- | --- | --- | --- | --- |
| Constant Head Boundary Cells | |  | Constant Head Boundary Cells | |
| Row | Column |  | Row | Column |
| 1 | 1 |  | 13 | 29 |
| 1 | 2 |  | 14 | 30 |
| 1 | 3 |  | 15 | 31 |
| 1 | 4 |  | 16 | 32 |
| 1 | 5 |  | 17 | 33 |
| 1 | 6 |  | 18 | 33 |
| 1 | 7 |  | 19 | 34 |
| 1 | 8 |  | 20 | 34 |
| 2 | 9 |  | 21 | 35 |
| 2 | 10 |  | 22 | 36 |
| 2 | 11 |  | 23 | 36 |
| 2 | 12 |  | 24 | 37 |
| 3 | 13 |  | 25 | 37 |
| 3 | 14 |  | 26 | 38 |
| 3 | 15 |  | 27 | 38 |
| 4 | 16 |  | 28 | 38 |
| 4 | 17 |  | 29 | 39 |
| 5 | 18 |  | 30 | 39 |
| 5 | 19 |  | 31 | 39 |
| 6 | 20 |  | 32 | 39 |
| 7 | 21 |  | 33 | 40 |
| 7 | 22 |  | 34 | 40 |
| 8 | 23 |  | 35 | 40 |
| 9 | 24 |  | 36 | 40 |
| 9 | 25 |  | 37 | 40 |
| 10 | 26 |  | 38 | 40 |
| 11 | 27 |  | 39 | 40 |
| 12 | 28 |  | 40 | 40 |

There are several sources of error in this test case. The first is that a different version of MODFLOW was used to calculate the underlying flow model. The second is that the particles were calculating using a method very similar to that used by MODPATH, but did not use the version of MODPATH tested in this document. The third is that the locations of the points at 0, 2,500, 5,000, and 7,500 days were not explicitly stated and had to be reproduced. Errors may have been introduced in the reproduction, either through the calculation of the location of the starting points, or through minor errors in the digitization of Figure 2‑3.

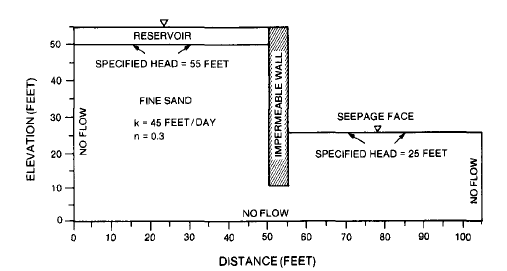
The acceptance criteria for this test are that the MODPATH simulation are as follows:

* Criterion 1 – the MODPATH simulation produces straight particle tracks that radiate outward
* Criterion 2 – the percent difference between the length of the flow paths digitized from Figure 2‑3 calculated value must not be more than [5%]. The length of the flow paths calculated by MODPATH are calculated in post-processing by selecting the global X and Y values for each particle at the time of interest, and calculating the distance from that particle to the lower left corner of the model.

Criterion 1 gives evidence that the particle tracks are moving according to the flow direction. Criterion 2 gives evidence that the particles are moving at a speed that is within an acceptable range of similarity to the particles in Pollock, 1988. Criterion 1 must be determined using professional judgment, by observing the image files in Test\_Case\_1/output/figures to confirm that the particle tracks radiate outward, as seen in [Figure 2-3]. The pass/fail status of criterion 2 will be printed to “pass\_fail.csv” [in the root directory], and detailed results are printed in Test\_Case\_1/output “tc1\_results.csv”.

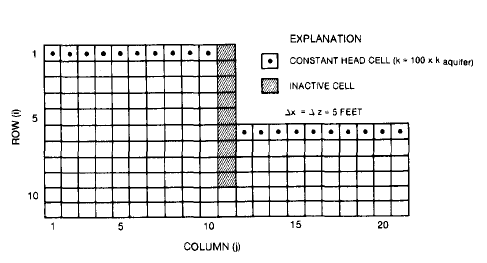
***Test Case 2: Particle Tracking in Non-Uniform Flow***

This test problem for MODPATH involves examining flow under an impermeable wall. This test is repeated from and compared against a test in Pollock, 1988. The second test involves flow under an impermeable wall (Figure 2‑4). To the left of an impermeable retaining wall is a 50-foot thickness of fine sand overlain by 5 feet of water in a reservoir. A seepage face at an elevation of 25 feet above the base of the aquifer is present to the right of the retaining wall. Figure 2‑5 shows the boundary conditions in the finite-difference grid used to simulate constant heads of 55 feet to the left of the wall and 25 feet to the right, and the no-flow cells representing the wall. Particles were released at the base of the reservoir and tracked forward until they reached the seepage face (Figure 2‑6). The particle tracks in [Figure 2-6] were digitized. This test is based on comparing the digitized particle tracks from [Figure 2-6] against the MODPATH particle tracks created by recreating the model from Pollock, 1988.



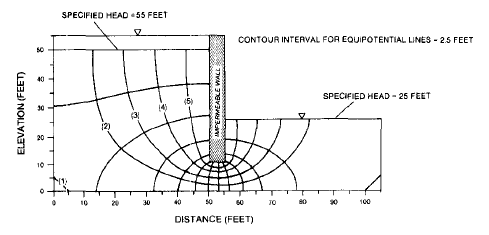
Source: Pollock, 1988

Figure 2‑4. Test Case 2 Layout



Source: Pollock, 1988

Figure 2‑5. Test Case 2 MODFLOW Layout



Source: Pollock, 1988

Figure 2‑6. Particle Location Over Time in Test Case 2

This model has a single layer, with 11 rows and 21 columns, each with a length and width of 5 ft. The thickness of the model is set to a uniform 25 feet. The thickness was not defined in Pollock, 1988, so a thickness of 25 feet was assumed. No-flow cells were assigned as seen in Figure 2‑5. Constant heads of 55 and 25 feet were assigned to the cells representing the reservoir and seepage face, respectively, as seen in Figure 2‑5. Starting heads were also not listed in Pollock, 1998, so it was assumed that the area beneath the reservoir and the impermeable berm had a starting head of 55, and the area beneath the seepage face had a starting head of 25. The particle release locations seen in Table 2‑5 were estimated based on Figure 2‑6. Particle 1 was not modeled.

The python script which made the MODFLOW and MODPATH inputs and executed the MODFLOW and MODPATH simulations is [“S01\_build\_pollock\_88\_ex2.py”]. Starting particle locations were calculated in “Write\_starting\_locations.py”. This script also executed the post-processing [get info on what the post-processed stuff is called and where it is, and put it here]. The model files are stored in /workspace, and post-processed model results are stored in [/output. Within /output, /figures contains outputs of figures of the model results at 2,500, 5,000, and 7,500 days. The file “tc1\_results.csv” contains the information used to determine the pass or fail status of the test.]

S00\_pngs\_2\_tiffs.py converted the image file of [Figure whatever] to a raster so the flow lines could be digitized. The image file, Figure\_10\_flownet.png is saved in Test\_Case\_2/gwpath\_images, and the resulting raster, Figure\_10\_flownet.tif, is saved in Test\_Case\_2/gwpath\_rasters, along with the georeferenced version of that raster, Figure\_10\_referenced, and a shapefile, starting\_location.shp, used in creating the raster. The folder/digitize contains two shapefiles: figure\_10\_flownet.shp, the shapefile digitized from Figure\_10\_referenced, and endpoints.shp, which contain the approximate ending locations of the particle tracks from figure\_10\_referenced.

| Table 2‑4. Model Properties for Test Case 2 | | | |
| --- | --- | --- | --- |
| Variable name | Variable value | Units | Source |
| Number of rows | 11 | N/A | Pollock, 1988, Figure 9 |
| Number of columns | 21 | N/A | Pollock, 1988, Figure 9 |
| Height of rows | 5 | ft | Pollock, 1988, Figure 9 |
| Width of columns | 5 | ft | Pollock, 1988, Figure 9 |
| Number of layers | 1 | N/A | Assumed |
| Thickness of layer | 25 | ft | Assumed |
| Hydraulic conductivity (aquifer) | 45 | ft/day | Pollock, 1988, Figure 8 |
| Hydraulic conductivity (reservoir and seepage face) | 4,500 | ft/day | Pollock, 1988, Figure 9 |
| Porosity | 0.3 | N/A | Pollock, 1988, Figure 8 |
| Constant head (reservoir) | 55 | ft | Pollock, 1988, Figure 8 |
| Constant head (seepage face) | 25 | ft | Pollock, 1988, Figure 8 |
| Starting head (columns 1 through 11) | 55 | ft | Assumed |
| Starting head (columns 12 through 21) | 25 | ft | Assumed |
| Number of particles | 5 | N/A | Pollock, 1988, Figure 10 |
| Number of stress periods | 30 | N/A | Assumed |
| Stress period length | 1 | days | Assumed |

There are several sources of error in this test case. The first is that a different version of MODFLOW was used to calculate the underlying flow model. The second is that the particles were calculating using a method very similar to that used by MODPATH, but did not use the version of MODPATH tested in this document. The third is that [Figure 2-6] was slightly warped and required georeferencing to reduce the effect on the pathline shape. Some effect of the warped figure may still remain on the digitized pathlines, causing a slightly different shape than the true results.

The acceptance criteria for this test are that the MODPATH simulation are as follows:

* Criterion 1 – the MODPATH simulation produces particle tracks that are visually similar to those seen in [Figure 2-6], according to professional judgment.
* Criterion 2 – the percent difference between the particle end points digitized from [Figure 2-6] and the MODPATH-calculated particle end points must not be more than 10%.

Criterion 1 gives evidence that the particle tracks are moving according to the flow direction. Criterion 2 gives evidence that the particles’ ending locations are within an acceptable range of similarity to the particles in Pollock, 1988. Criterion 1 must be determined using professional judgment, by observing the image files in [Test\_Case\_2/output/figures] to confirm that the particle tracks follow pathlines similar to those seen in [Figure 2-3]. The pass/fail status of criterion 2 will be printed to “pass\_fail.csv” [in the root directory], and detailed results are printed in [Test\_Case\_2/output “tc1\_results.csv”].

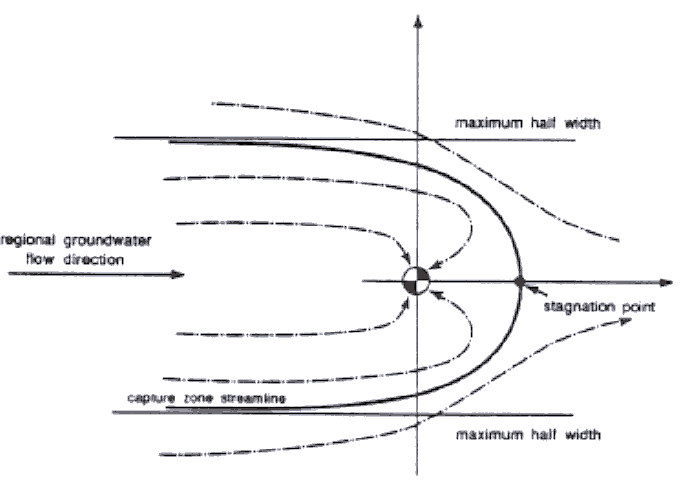
***Test Case 3: Capture Zone in an Isotropic Unconfined Aquifer***

This test problem for MODPATH involves defining the shape of the capture zone of a constantly-pumping extraction well in an isotropic, unconfined aquifer with a clear hydraulic gradient. An extraction well pumping water from an aquifer will only extract water from its capture zone. MODPATH’s reverse particle tracking capabilities are commonly used to map capture zones in an unconfined aquifer at the Hanford Site. In this test, results from a MODPATH reverse particle tracking simulation mapping the shape of a capture zone are compared to an analytical solution describing elements of the shape of that capture zone. The analytical solution calculates the maximum width of a capture zone and stagnation point for an extraction well in an unconfined aquifer in steady-state conditions (Grubb, 1993, *Analytical Model for Estimation of Steady-State Capture Zones of Pumping Wells in Confined and Unconfined Aquifers*). MODPATH particle tracks were compared to the analytical maximum width of the aquifer and location of the stagnation point.

Grubb’s (1993) analytical model for defining the shape of an unconfined capture zone assumes the following:

* The aquifer is homogeneous, isotropic, and infinite in horizontal extent.
* The flow conditions are uniform and steady-state.
* The unconfined aquifer has a horizontal lower confining layer with no leakage, rainfall infiltration, or other vertical recharge.
* The specific yield of the aquifer is neglected.
* Vertical gradients are negligible.
* The pumping well is fully penetrating, open over the thickness of the aquifer, and pumps at a constant rate.

Taking all these assumptions into account, a well pumping continuously at a constant rate will produce a capture zone extend infinitely in the upgradient direction and end in the downgradient direction, as shown in Figure 2‑1. The maximum width of the upgradient capture zone as measured from the center of the well (ymax­) and the stagnation point, the distance from the pumping well to the down-gradient edge of the capture zone (the stagnation point, or *x0*), can be calculated analytically.



Source: Yang, Spencer, and Gates, 1995, *Analytical Solutions for Determination of Non-Steady-State and Steady-State Capture*

Figure 2‑4. Capture zone in an unconfined aquifer

The maximum width of a capture zone an unconfined aquifer is defined by the following equation from Grubb (1993):

where:

*ymax* = the maximum width of the capture zone, where y0 intersects the center of the well

*Q* = the pumping rate

*L* = the distance between *h1* and *h2*

*K* = the hydraulic conductivity

*h1* = the upgradient head

*h2* = the downgradient head

The position of the stagnation point is calculated using the following equation:

where:

*x0* = the stagnation point

*Q* = the pumping rate

*L* = the distance between *h1* and *h2*

*K* = the hydraulic conductivity

*h1* = the upgradient head

*h2* = the downgradient head

The values used to calculate the analytical solution are shown in Table 1. The extraction rate (Q) used is a value similar to those use in remediation at the Hanford site, the hydraulic conductivity (K) falls within the range of the K seen in the Ringold Wooded Island Formation Member unit E (CP-47631, *Model Package Report: Central Plateau Groundwater Model*), and the gradient also falls within the range of gradients seen at the Hanford site (ECF-Hanford-17-0241, *Hydraulic Gradient and Velocity Calculations for RCRA Sites in 2017*). The MODFLOW model used the Q and K shown in Table 1. The left and right constant head cell values were set to h1 and h2, respectively. The width of the model was set to 80,000 meters, to get as close to an infinite aquifer as possible while avoiding overly long computational time.

The python script which made the MODFLOW inputs and executed the MODFLOW simulations is [“S01\_50\_ft.py”]. The python script which made the MODPATH inputs and executed the MODPATH simulations is [“S02\_50\_ft\_mp.py”]. The script which executed the post-processing is [S03\_pp\_tc3.py]. The script fetter.py contains the functions which calculate the ymax and the x0; this script is called during post-processing. The model files are stored in /workspace, and post-processed model results are stored in /output. Within /output, there are two files: backwards.png and “percent\_diff\_ymax\_xstag.csv”. The file “percent\_diff\_ymax\_xstag.csv” contains the information used to determine the pass or fail status of the test.

| Table 2‑1. Model Properties Used in Analytical Solution | | |
| --- | --- | --- |
| Variable name | Variable Value | Units |
| Extraction rate (Q) | 100 (19,455) | gal/min (cubic feet/day) |
| Hydraulic conductivity (K) | 1000 | ft/day |
| h1 head | 200 | ft |
| h2 head | 199.048 | ft |
| Distance between h1 and h2 (L) | 80,000 | ft |

| Table 2‑2. Model Properties | | |
| --- | --- | --- |
| **Variable name** | **Variable Value** | **Units** |
| Pumping well location | Row 401, column 1067 | N/A |
| Porosity | 0.3 | N/A |
| Constant head (left) | 200 | ft |
| Constant head (right) | 167.35 | ft |
| Number of layers | 1 | N/A |
| Starting head | 200 | ft |

Reverse particle tracking was chosen to evaluate the capture zone shape. Two particles were seeded at the right boundary of the well cell, one at 49.9% and one at 50.1% of the cell length in the y direction. This was done to get the particle tracks of the maximum outer boundary of the capture zone as mappable using reverse particle tracking. In the z direction, the particles were both seeded at 50% of the cell height.

The ymax and x0 values of the capture zone created by the MODPATH particle tracks were checked using a python script, S03\_ymax\_and\_stagnation\_point.py. To evaluate ymax, this script retrieved the final global Y value from the endpoint file. After subtracting the y value for the midpoint of the well cell, the percent difference from the analytical ymax was calculated. The x0 was calculated in a similar manner: the script retrieved the maximum global X from the pathline file and calculated the percent difference from the analytical x0.

One source of error in this test case is that the maximum y extent on the southern particle track is slightly less than the maximum y extent on the northern particle track.

The analytical ymax is calculated to be 4,054 feet, and x0 is calculated to be 1,290 feet. The acceptance criteria for this test are that the MODPATH pathlines be within [10?] % of the analytical results. [The results are stored [here]].

# OPERATIONAL TESTING

The MODFLOW and MT3DMS software will be subjected to operational periodic testing under a graded approach. It is recognized that computational software of this type will only change in performance under three possible conditions:

1. hardware change
2. computational software change
3. operating system (environment) change

The first type of change (in hosting hardware) is already addressed through the requirement for software installation and checkout (Section 3.3, Test Cases) which must be repeated to qualify any new installation of the software on new hardware.

The second type of change (in the computational software itself) is already addressed by the requirement to perform all of the acceptance tests provided in this test plan, issue a revision to the software acceptance test report to document the results of this testing, and promote the new version to production use.

The third type of change represents the one condition that must be monitored by users. Here, even patches to the operating system constitute a change to the configuration of the operating system. Accordingly, users are required to re-test their installation using Test Cases MF-ITC-1 and MT-ITC-1 before use for decision-informing calculations if their operating system configuration has changed since the last installation test was checked. Performance of this operational testing is assigned to software users for single-user systems (e.g., Windows™ workstations that are subject to frequently, even weekly, operating system patches), but will be performed by the software owner or designee for multi-user operating systems (e.g., Linux™ cluster systems that require only infrequent patches with advance notice by system administrators).

# REFERENCES

CP-47631, *Model Package Report: Central Plateau Groundwater Model*

ECF-Hanford-17-0241, *Hydraulic Gradient and Velocity Calculations for RCRA Sites in 2017* [*https://pdw.hanford.gov/arpir/pdf.cfm/viewDoc?accession=0066390H*](https://pdw.hanford.gov/arpir/pdf.cfm/viewDoc?accession=0066390H)*.*

Grubb, 1993, *Analytical Model for Estimation of Steady-State Capture Zones of Pumping Wells in Confined and Unconfined Aquifers*

**Acronyms, Abbreviations, and Definitions**

| Acronym | Description |
| --- | --- |
| ATR | Acceptance Test Report |
| CHPRC | CH2M HILL Plateau Remediation Company |
| FRD | Functional Requirements Document |
| MKS Integrity™ | Configuration management system software |
| MODFLOW | MODular three-dimensional finite-difference ground-water FLOW model (software) |
| MT3DMS | modular three-dimensional, multi-species transport model for simulation of advection, dispersion, and chemical reactions of contaminants in groundwater systems (software) |
| SMP | Software Management Plan |
| STP | Software Test Plan |
| USGS | U.S. Geological Survey |

# ATTACHMENTS

1. Template: Test Log for MP-ATC-1
2. Template: Test Log for MU-ATC-1

ATTACHMENT 1

Test Log for MODPATH Acceptance Test Case 1 (MP-ATC-1)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| MODFLOW Acceptance Test Case 1  Description: Forward Particle Tracking from an Injection Well | | | Test Case #: Mp-ATC-1 | | Date: | |
| System Attributes:  Version #:  Release #:  Environment:  Server: | | | Test Performed by: | | | |
| Test Step # | Requirement # | Test Instruction | | Expected Result | | Actual Result |
| 1 | Obtain source code or executable for MODPATH code from software owner & install on target computer | Compile using appropriate Fortran compiler, if necessary | | MODPATH executable is ready and functional | |  |
|  | Obtain source code or executable for MODFLOW-2000 code from software owner & install on target computer | Compile using appropriate Fortran compiler, if necessary | | MODFLOW-2000 executable is ready and functional | |  |
| 2 | [Download python] |  | |  | |  |
| 3 | [Setup the python environment] |  | |  | |  |
| 2 | Obtain files for test problem from software owner | Copy files to appropriate test directory | | Test files are ready for use | |  |
| 3 | Run MODFLOW to solve flow for test case | Execute MODFLOW against [.nam] name file in test directory | | MODFLOW executes without error | |  |
|  | Run MODPATH to solve particle tracking for test case | Execute MODPATH against [.mpnam] name file in test directory | | MODPATH executes without error | |  |
| 5 | Use test results and graphics from spreadsheet to complete test reporting in ATR | Copy and paste graphics and note acceptance test results in ATR | | MODFLOW acceptance test criterion are met | |  |

ATTACHMENT 1

Test Log for MODPATH Acceptance Test Case 1 (MU-ATC-1)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| MODPATH Acceptance Test Case 1  Description: Theis Transient Drawdown Problem | | | Test Case #: MF-ATC-1 | | Date: | |
| System Attributes:  Version #:  Release #:  Environment:  Server: | | | Test Performed by: | | | |
| Test Step # | Requirement # | Test Instruction | | Expected Result | | Actual Result |
| 1 | Obtain source code or executable for MODFLOW-2000-MST code from software owner & install on target computer | Compile using appropriate Fortran compiler, if necessary | | MODFLOW executable is ready and functional | |  |
| 2 | Obtain files for test problem from software owner | Copy files to appropriate test directory | | Test files are ready for use | |  |
| 3 | Run MODFLOW to solve for flow problem | Execute MODFLOW against mf-atc-1.nam name file in test directory | | MODFLOW executes without error | |  |
| 4 | Extract results and transfer to validation spreadsheet | Using a text editor, copy MODFLOW-calculated drawdown values from end of list file mf-atc-1.lis and paste into appropriate cells in validation spreadsheet “mt-atc-1.xlsx” | | Spreadsheet will update tables, graphics, and acceptance test results | |  |
| 5 | Use test results and graphics from spreadsheet to complete test reporting in ATR | Copy and paste graphics and note acceptance test results in ATR | | MODFLOW acceptance test criterion are met | |  |

ATTACHMENT 2

Test Log for MT3DMS Acceptance Test Case 1 (MT-ATC-1)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| MT3DMS Acceptance Test Case 1  Description: One-Dimensional Advective-Diffusive Transport | | | Test Case #: MT-ATC-1 | | Date: | |
| System Attributes:  Version #:  Release #:  Environment:  Server: | | | Test Performed by: | | | |
| Test Step # | Requirement # | Test Instruction | | Expected Result | | Actual Result |
| 1 | Obtain source code or executable for MODFLOW-2000-MST and MT3DMS code from software owner & install on target computer | Compile using appropriate Fortran compiler, if necessary | | MODFLOW and MT3DMS executables are ready and functional | |  |
| 2 | Obtain files for test problem from software owner | Copy files to appropriate test directory | | Test files are ready for use | |  |
| 3 | Run MODFLOW to solve for flow problem | Execute MODFLOW against mt-atc-1.nam name file in /flow test directory | | MODFLOW executes without error; flow results transfer file is ready for use in next step | |  |
| 4 | Run MT3DMS to solve for transport in both time durations | Execute MT3DMS against mt-atc-1.nam name file in both /transport\_2400d and /transport\_9600d subdirectories | | MT3DMS executes without error; transport results are available in mt-atc-1.m3d in both time solution subdirectories | |  |
| 5 | Extract results and transfer to validation spreadsheets | Using a text editor, copy MT3DMS-calculated concentration values from end of list file mf-atc-1.m3d and paste into appropriate cells in validation spreadsheets “mt-atc-1\_2400d.xlsx” and “mt-atc-1\_9600d.xlsx” | | Spreadsheets will update tables, graphics, and acceptance test results | |  |
| 6 | Use test results and graphics from spreadsheet to complete test reporting in ATR | Copy and paste graphics and note acceptance test results in ATR | | MT3DMS acceptance test criterion are met | |  |

1. MKS is a trademark of MKS, Inc. [↑](#footnote-ref-1)