

**RocFall3**

# **Runout Distances**

Unit Test Verification Manual

# Table of Contents

<b>RocFall3 Runout Distances Unit Tests .....</b>	<b>3</b>
<b>Background .....</b>	<b>4</b>
<b>1. Unit Test Set #1: Rock trajectory crossing a Polyline .....</b>	<b>5</b>
<b>2. Unit Test Set #2: Rock trajectory not crossing a Polyline .....</b>	<b>6</b>
<b>3. Unit Test Set #3: Rock trajectory wrapping around a polyline without intersection .....</b>	<b>7</b>
<b>4. Unit Test Set #4: Rock trajectory wrapping around a polyline with intersection.....</b>	<b>8</b>
<b>5. Unit Test Set #5: Rock trajectory starting on polyline .....</b>	<b>9</b>
<b>6. Unit Test Set #6: Rock trajectory starting on the undefined trajectory of a polyline .....</b>	<b>10</b>
<b>7. Unit Test Set #7: Rock trajectory crossing a polyline.....</b>	<b>11</b>
<b>8. Unit Test Set #8: Rock trajectory not crossing a polyline .....</b>	<b>12</b>
<b>9. Unit Test Set #9: Rock trajectory wrapping around a polyline without intersection .....</b>	<b>13</b>
<b>10. Unit Test Set #10: Rock trajectory crossing a barrier .....</b>	<b>14</b>
<b>11. Unit Test Set #11: Rock trajectory not crossing a barrier .....</b>	<b>15</b>
<b>12. Unit Test Set #12: Rock trajectory wrapping around a barrier without intersection .....</b>	<b>16</b>

# RocFall3 Runout Distances Unit Tests

---

This document presents the unit tests that have been used to verify the accuracy of the reference runout distance feature in RocFall3. RocFall3 is a 3D engineering analysis program for assessing rockfall risks in rock slopes, produced by Rocscience Inc. of Toronto, Canada. The purpose of this verification is to compare program outputs to manually computed analytical results to ensure RocFall3 reference runout distance results are accurate for both lumped mass and rigid body methods.

# Background

---

The reference runout distance feature in RocFall3 calculates the distance between the endpoint of a rock and the point where the rock's trajectory *first* intersects a reference object (polyline or barrier). RocFall3 offers three runout distance results, their respective definitions and formulations are presented below:

$$\text{Runout Total (3D distance)} = \sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2}$$

$$\text{Runout XY (distance in the 2D XY plane)} = \sqrt{(x - x_0)^2 + (y - y_0)^2}$$

$$\text{Runout Z (distance in the z dimension)} = |z - z_0|$$

Where  $(x, y, z)$  defines the endpoint of a rock and  $(x_0, y_0, z_0)$  defines the first point of intersection between a rock and a reference.

Using a polyline, the point of intersection is calculated as the intersection in the XY plane between the polyline and the rock's trajectory with the Z-coordinate taken as the Z-coordinate of the rock at the point of intersection.

If no intersection is found between a rock's trajectory and a reference object, no runout distances are returned.

In the subsequent sections, unit tests are presented for both rigid body and lumped mass methods. A polyline was used as the reference object for both methods and barriers are tested as well. Below is a summary of both seeder and material properties used in the unit tests.

## Property Set #1

Seeder properties: using a cube shape, with 2700 kg mass, and 2700 kg/m<sup>3</sup> density.

Material properties: 0.5 normal restitution and dynamic friction.

## Property Set #2

Seeder properties: using an icosphere shape, with 1000 kg mass, and 2700 kg/m<sup>3</sup> density.

Material properties: 0.35 normal restitution and 0.5 dynamic friction.

**For all unit tests, project settings were left as default values and all statistical distributions were set to "None". All data presented in this verification manual follow metric units (m, kg, kJ).**

# 1. Unit Test Set #1: Rock trajectory crossing a Polyline

This unit test aims to test the basic functionality of the runout distance calculation when a rock crosses the polyline from either side.

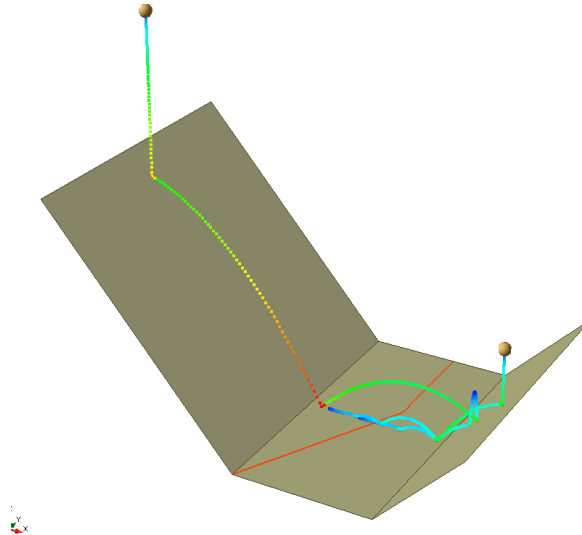
This unit uses Property Set #1 and the following geometries:

Seeder 1: (0.8, 5, 20). Seeder 2: (14, 5, 11)

Polyline: (10, 7, 7), (10, 6, 7), (7, 0, 7)

Slope: (0, 0, 15), (7, 0, 7), (12, 0, 7), (15, 0, 10). The line defined by these points is extruded by 10 m in the Y-direction to form the surface.

Seeders have no initial velocities.



**Table 1-1: Seeder 1 Results**

Intersection Point (m)	(9.50, 5.00, 9.02)
Endpoint (m)	(10.31, 5.00, 7.50)

**Table 1-2: Seeder 1 Runout Distances**

	Analytical Result (m)	Program Output (m)
Runout Total	1.72	1.725
Runout XY	0.81	0.808
Runout Z	1.52	1.524

**Table 1-3: Seeder 2 Results**

Intersection Point (m)	(9.50, 5.00, 7.61)
Endpoint (m)	(9.35, 5.00, 7.50)

**Table 1-4: Seeder 2 Runout Distances**

	Analytical Result (m)	Program Output (m)
Runout Total	0.19	0.191
Runout XY	0.15	0.154
Runout Z	0.11	0.113

This unit test aims to test the basic functionality of the runout distance calculation when a rock crosses the polyline from either side. Seeder 1's results further show that only the first intersection is considered in the calculation.

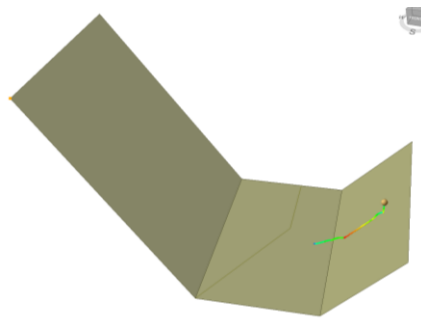
Intersections at a vertex also produce the same results.

## 2. Unit Test Set #2: Rock trajectory not crossing a Polyline

---

This unit test aims to test the output of the program when a rock does not cross the polyline.

This test uses the same properties and geometries as unit test #1 except with the seeder defined at (14, 5, 9.5).



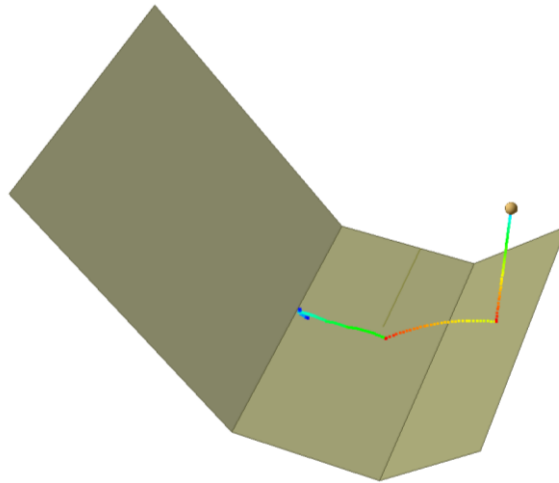
In this unit test, there are no intersections, and the program correctly returns no runout distances.

### 3. Unit Test Set #3: Rock trajectory wrapping around a polyline without intersection

---

This unit test aims to ensure the program does not detect an intersection if a rock's trajectory wraps around a polyline.

This test uses the same properties and geometries as unit test #1 except with the seeder defined at (14, 5, 15) and the first segment of the polyline is deleted.



In this unit test, there are no intersections, and the program correctly returns no runout distances.

## 4. Unit Test Set #4: Rock trajectory wrapping around a polyline with intersection

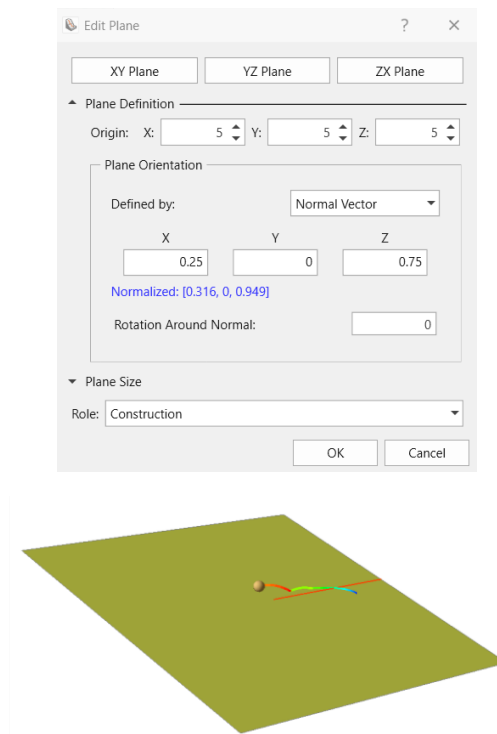
This unit test aims to ensure that rocks that wrap around a polyline and then intersect are properly calculated.

This unit test uses property set two and the following geometries:

Seeder: (4.5, 5, 6) with translational velocity of 7 m/s and trend of 350°.

Slope: an XY plane around the origin with the normal vector in the (0.25, 0, 0.75) direction before normalization.

Polyline: (4.5, 14, 5.166) (4.5, 6, 5.166) on the slope surface



**Table 4-1: Seeder 1 Results**

Intersection Point (m)	(4.50, 10.35, 5.17)
Endpoint (m)	(5.36, 11.25, 4.88)

**Table 4-2: Seeder 1 Runout Distances**

	Analytical Result (m)	Program Output (m)
Runout Total	1.27	1.273
Runout XY	1.24	1.241
Runout Z	0.29	0.282

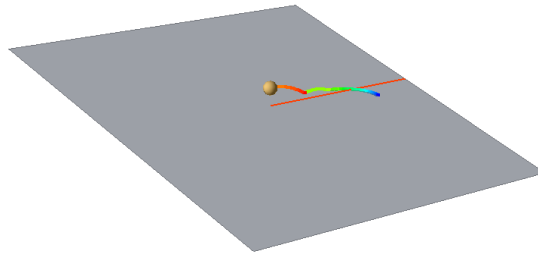


## 5. Unit Test Set #5: Rock trajectory starting on polyline

---

This unit test aims to ensure that rocks starting on a polyline have the initial point counted as an intersection.

This test uses the same properties and geometries as unit test #4 but with the polyline extended past the seeder (Y value less than or equal to 5).



**Table 5-1: Seeder 1 Results**

Intersection Point (m)	(4.50, 10.35, 5.17)
Endpoint (m)	(5.36, 11.25, 4.88)

**Table 5-2: Seeder 1 Runout Distances**

	Analytical Result (m)	Program Output (m)
Runout Total	6.41	6.403
Runout XY	6.31	6.305
Runout Z	1.12	1.119

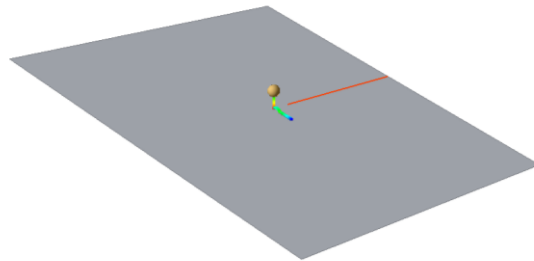
Placing the seeder on an edge versus on a vertex produces the same results.

## 6. Unit Test Set #6: Rock trajectory starting on the undefined trajectory of a polyline

---

This unit test aims to ensure the end of detection of a polyline is correct. A seeder is placed on the same axis as the end segment of a polyline but not on the line itself, which should not register as an intersection.

This test uses the same properties and geometries as unit test #4 but with 0 initial velocity.



In this unit test, there are no intersections, and the program correctly returns no runout distances.

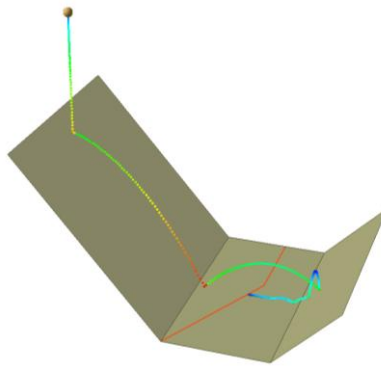
In the following unit tests, only basic functionality will be presented.

## 7. Unit Test Set #7: Rock trajectory crossing a polyline

---

This unit test aims to test the basic functionality of the runout distance calculation when a rock crosses the polyline.

This test uses the same properties and geometries as unit test #1 without seeder 2 and with lumped mass method chosen in project settings.



**Table 7-1: Seeder 1 Results**

Intersection Point (m)	(9.50, 5.00, 8.61)
Endpoint (m)	(9.59, 5.00, 7.00)

**Table 7-2: Seeder 1 Runout Distances**

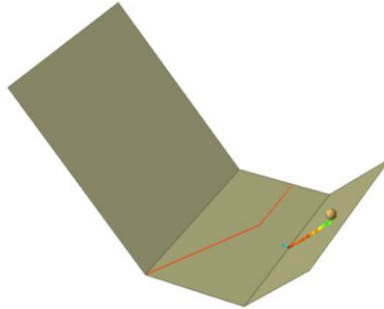
	Analytical Result (m)	Program Output (m)
Runout Total	1.61	1.608
Runout XY	0.09	0.088
Runout Z	1.61	1.606

## 8. Unit Test Set #8: Rock trajectory not crossing a polyline

---

This unit test aims to test the output of the program when a rock does not cross the polyline.

This test uses the same properties and geometries as unit test #7 except with the seeder defined at (14, 5, 9.5)



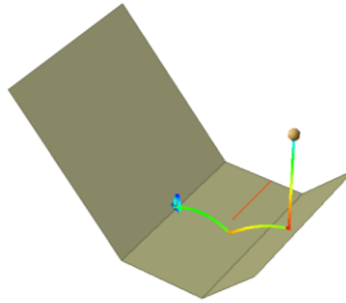
In this unit test, there are no intersections, and the program correctly returns no runout distances.

## 9. Unit Test Set #9: Rock trajectory wrapping around a polyline without intersection

---

This unit test aims to ensure the program does not detect an intersection if a rock's trajectory wraps around a polyline.

This test uses the same properties and geometries as unit test #7 except with the seeder defined at (14, 5, 15) and the first segment of the polyline is deleted.



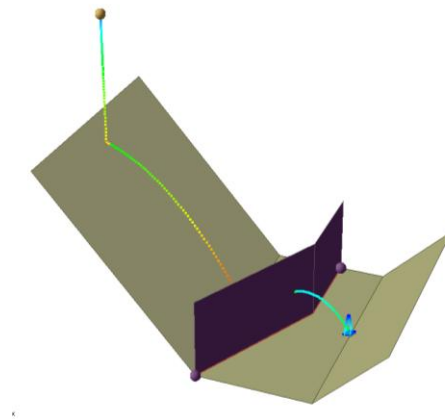
In this unit test, there are no intersections, and the program correctly returns no runout distances.

## 10. Unit Test Set #10: Rock trajectory crossing a barrier

---

This unit test aims to test the basic functionality of the runout distance calculation when a rock crosses the barrier.

This test uses the same properties and geometries as unit test #4. The barrier is defined at the same location as the polyline with a height of 3.5 m and a capacity of 25 kJ.



**Table 10-1: Seeder 1 Results**

Intersection Point (m)	(9.50, 5.00, 8.61)
Endpoint (m)	(11.71, 5.00, 7.00)

**Table 10-2: Seeder 1 Runout Distances**

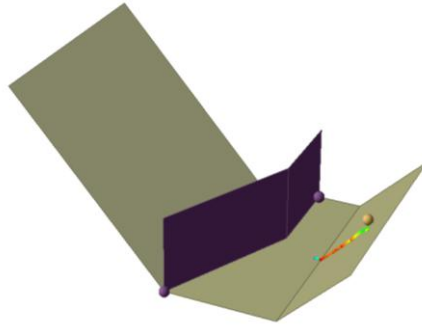
	Analytical Result (m)	Program Output (m)
Runout Total	2.73	2.731
Runout XY	2.21	2.209
Runout Z	1.61	1.606

## 11. Unit Test Set #11: Rock trajectory not crossing a barrier

---

This unit test aims to test the output of the program when a rock does not cross the barrier.

This test uses the same properties and geometries as unit test #10 except with the seeder defined at (14, 5, 9.5)



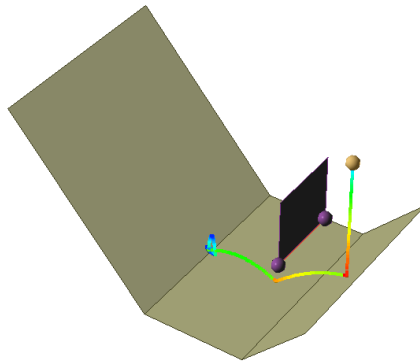
In this unit test, there are no intersections, and the program correctly returns no runout distances.

## 12. Unit Test Set #12: Rock trajectory wrapping around a barrier without intersection

---

This unit test aims to ensure the program does not detect an intersection if a rock's trajectory wraps around a barrier.

This test uses the same properties and geometries as unit test #10 except with the seeder defined at (14, 5, 15) and the first section of the barrier is deleted.



In this unit test, there are no intersections, and the program correctly returns no runout distances.