# Matrix Multiplication — Final Report (v3.4) — AI-generated tests only

Generated: 2025-11-13 12:11:28Z

This v3.4 report contains only AI-generated challenging test cases (no user literals). Rules applied consistently: - Do NOT zero-fill missing values. - Use matching element pairs for dot-products; continue if surplus exists. - Mark cells null when insufficient; strict expected marks Invalid/Discarded if any required grid element missing. Each test shows: Java literal, Java-like List, My-code result (simulated), Expected strict result (discard-on-missing), and notes.

## *Test #1 (synthetic_1001)*

Java literal (verbatim):

```
Integer[][][] test = new Integer[][][] {{{1,2,3},{4,5,6},{7,8,9}},{{9,8,7},{6,5,4},{3,2,1}}};
```

Actual Java Lists executed:

```
[
  [[1, 2, 3], [4, 5, 6], [7, 8, 9]],
  [[9, 8, 7], [6, 5, 4], [3, 2, 1]]
]
```

System.out.println outputs (simulated mapping - none captured):

```
(none)
```

Step 1: Multiply A(3, 3) x B(3, 3)

```
  My-code result: [
    [30, 24, 18],
    [84, 69, 54],
    [138, 114, 90]
  ]
  Expected (strict): [
    [30, 24, 18],
    [84, 69, 54],
    [138, 114, 90]
  ]
```

### *Test #2 (synthetic_1002)*

Java literal (verbatim):
```
Integer[][][] test = new Integer[][][] {{{1},{2,3},{4,5,6}},{{6,7},{8,9},{10,11}}};
```

Actual Java Lists executed:
```
[
  [[1], [2, 3], [4, 5, 6]],
  [[6, 7], [8, 9], [10, 11]]
]
```

System.out.println outputs (simulated mapping - none captured):
```
(none)
```

Step 1: Multiply A(3, 3) x B(3, 2)
```
  My-code result: [
    [6, 7],
    [36, 41],
    [124, 139]
  ]
  Expected (strict): Invalid/Discarded (Invalid/Discarded due to missing A element)
```

## *Test #3 (synthetic_1003)*

Java literal (verbatim):
```
Integer[][][] test = new Integer[][][] {{{1,2}},{{3},{4}},{{5,6}},{{7},{8}}};
```

Actual Java Lists executed:
```
[
  [[1, 2]],
  [[3], [4]],
  [[5, 6]],
  [[7], [8]]
]
```

System.out.println outputs (simulated mapping - none captured):
```
(none)
```

Step 1: Multiply A(1, 2) x B(2, 1)
```
  My-code result: [
    [11]
  ]
  Expected (strict): [
    [11]
  ]
```

Step 2: Multiply A(1, 1) x B(1, 2)
```
  My-code result: [
    [55, 66]
  ]
  Expected (strict): [
    [55, 66]
  ]
```

Step 3: Multiply A(1, 2) x B(2, 1)
```
  My-code result: [
    [913]
  ]
  Expected (strict): [
    [913]
  ]
```

## *Test #4 (synthetic_1004)*

Java literal (verbatim):

```
Integer[][][] test = new Integer[][][] {{{1,2,3,4}},{{5},{6},{7},{8}}};
```

Actual Java Lists executed:

```
[
  [[1, 2, 3, 4]],
  [[5], [6], [7], [8]]
]
```

System.out.println outputs (simulated mapping - none captured):

```
(none)
```

Step 1: Multiply A(1, 4) x B(4, 1)

```
  My-code result: [
    [70]
  ]
  Expected (strict): [
    [70]
  ]
```

## *Test #5 (synthetic_1005)*

Java literal (verbatim):

```
Integer[][][] test = new Integer[][][] {{{1,null,3},{4,5,null}},{{7,8},{9,10},{11,12}}};
```

Actual Java Lists executed:

```
[
  [[1, None, 3], [4, 5, None]],
  [[7, 8], [9, 10], [11, 12]]
]
```

System.out.println outputs (simulated mapping - none captured):

```
(none)
```

Step 1: Multiply A(2, 3) x B(3, 2)

```
  My-code result: [
    [40, 44],
    [73, 82]
  ]
  Expected (strict): Invalid/Discarded (Invalid/Discarded due to missing A element)
```

## *Test #6 (synthetic_1006)*

Java literal (verbatim):
```
Integer[][][] test = new Integer[][][] {{{}},{{1,2},{3,4}}};
```

Actual Java Lists executed:
```
[
  [[]],
  [[1, 2], [3, 4]]
]
```

System.out.println outputs (simulated mapping - none captured):
```
(none)
```

Step 1: Multiply A(1, 0) x B(2, 2)
```
  My-code result: [
    [0, 0]
  ]
  Expected (strict): Invalid/Discarded (Incompatible dims colsA=0 rowsB=2)
```

## *Test #7 (synthetic_1007)*

Java literal (verbatim):

```
Integer[][][] test = new Integer[][][] {{{1},{2},{3},{4}},{{1,2,3,4}}};
```

Actual Java Lists executed:

```
[
  [[1], [2], [3], [4]],
  [[1, 2, 3, 4]]
]
```

System.out.println outputs (simulated mapping - none captured):

```
(none)
```

Step 1: Multiply A(4, 1) x B(1, 4)

```
  My-code result: [
    [1, 2, 3, 4],
    [2, 4, 6, 8],
    [3, 6, 9, 12],
    [4, 8, 12, 16]
  ]
  Expected (strict): [
    [1, 2, 3, 4],
    [2, 4, 6, 8],
    [3, 6, 9, 12],
    [4, 8, 12, 16]
  ]
```

## *Test #8 (synthetic_1008)*

Java literal (verbatim):
```
Integer[][][] test = new Integer[][][] {{{1}},{{2,3}},{{4},{5},{6}},{{7,8,9}}};
```

Actual Java Lists executed:
```
[
  [[1]],
  [[2, 3]],
  [[4], [5], [6]],
  [[7, 8, 9]]
]
```

System.out.println outputs (simulated mapping - none captured):
```
(none)
```

Step 1: Multiply A(1, 1) x B(1, 2)
```
  My-code result: [
    [2, 3]
  ]
  Expected (strict): [
    [2, 3]
  ]
```

Step 2: Multiply A(1, 2) x B(3, 1)
```
  My-code result: [
    [23]
  ]
  Expected (strict): Invalid/Discarded (Incompatible dims colsA=2 rowsB=3)
```

Step 3: Multiply A(1, 1) x B(1, 3)
```
  My-code result: [
    [161, 184, 207]
  ]
  Expected (strict): [
    [161, 184, 207]
  ]
```

## *Test #9 (synthetic_1009)*

Java literal (verbatim):
```
Integer[][][] test = new Integer[][][] {{{1,2},{3,4}},{{1,2},{3,4}},{{1,2},{3,4}}};
```

Actual Java Lists executed:
```
[
  [[1, 2], [3, 4]],
  [[1, 2], [3, 4]],
  [[1, 2], [3, 4]]
]
```

System.out.println outputs (simulated mapping - none captured):
```
(none)
```

Step 1: Multiply A(2, 2) x B(2, 2)
```
  My-code result: [
    [7, 10],
    [15, 22]
  ]
  Expected (strict): [
    [7, 10],
    [15, 22]
  ]
```

Step 2: Multiply A(2, 2) x B(2, 2)
```
  My-code result: [
    [37, 54],
    [81, 118]
  ]
  Expected (strict): [
    [37, 54],
    [81, 118]
  ]
```

## *Test #10 (synthetic_1010)*

Java literal (verbatim):

```
Integer[][][] test = new Integer[][][] {{{1,2,3,4,5}},{{6},{7},{8},{9},{10}}};
```

Actual Java Lists executed:

```
[
  [[1, 2, 3, 4, 5]],
  [[6], [7], [8], [9], [10]]
]
```

System.out.println outputs (simulated mapping - none captured):

```
(none)
```

Step 1: Multiply A(1, 5) x B(5, 1)

```
  My-code result: [
    [130]
  ]
  Expected (strict): [
    [130]
  ]
```

## *Test #11 (synthetic_1011)*

Java literal (verbatim):
```
Integer[][][] test = new Integer[][][] {{{1,2},{3}},{{4,5,6}},{{7},{8,9}}};
```

Actual Java Lists executed:
```
[
  [[1, 2], [3]],
  [[4, 5, 6]],
  [[7], [8, 9]]
]
```

System.out.println outputs (simulated mapping - none captured):
```
(none)
```

Step 1: Multiply A(2, 2) x B(1, 3)
```
  My-code result: [
    [4, 5, 6],
    [12, 15, 18]
  ]
  Expected (strict): Invalid/Discarded (Incompatible dims colsA=2 rowsB=1)
```

Step 2: Multiply A(2, 3) x B(2, 2)
```
  My-code result: [
    [68, 45],
    [204, 135]
  ]
  Expected (strict): Invalid/Discarded (Incompatible dims colsA=3 rowsB=2)
```

## *Test #12 (synthetic_1012)*

Java literal (verbatim):
```
Integer[][][] test = new Integer[][][] {{{1}},{{2}},{{3}},{{4}},{{5}}};
```

Actual Java Lists executed:
```
[
  [[1]],
  [[2]],
  [[3]],
  [[4]],
  [[5]]
]
```

System.out.println outputs (simulated mapping - none captured):
```
(none)
```

Step 1: Multiply A(1, 1) x B(1, 1)
```
  My-code result: [
    [2]
  ]
  Expected (strict): [
    [2]
  ]
```

Step 2: Multiply A(1, 1) x B(1, 1)
```
  My-code result: [
    [6]
  ]
  Expected (strict): [
    [6]
  ]
```

Step 3: Multiply A(1, 1) x B(1, 1)
```
  My-code result: [
    [24]
  ]
  Expected (strict): [
    [24]
  ]
```

Step 4: Multiply A(1, 1) x B(1, 1)
```
  My-code result: [
    [120]
  ]
  Expected (strict): [
    [120]
  ]
```

## *Test #13 (synthetic_1013)*

Java literal (verbatim):

```
Integer[][][] test = new Integer[][][] {{{},{1,2}},{{3},{4,5}}};
```

Actual Java Lists executed:

```
[
  [[], [1, 2]],
  [[3], [4, 5]]
]
```

System.out.println outputs (simulated mapping - none captured):

```
(none)
```

Step 1: Multiply A(2, 2) x B(2, 2)

```
  My-code result: [
    [null, null],
    [11, 10]
  ]
  Expected (strict): Invalid/Discarded (Invalid/Discarded due to missing A element)
```

## *Test #14 (synthetic_1014)*

Java literal (verbatim):

```
Integer[][][] test = new Integer[][][] {{{1,2,3,4,5},{1,2,3,4,5},{1,2,3,4,5}},{{1,2},{3,4},{5,6},{7,8},{9,10}}};
```

Actual Java Lists executed:

```
[
  [[1, 2, 3, 4, 5], [1, 2, 3, 4, 5], [1, 2, 3, 4, 5]],
  [[1, 2], [3, 4], [5, 6], [7, 8], [9, 10]]
]
```

System.out.println outputs (simulated mapping - none captured):

```
(none)
```

Step 1: Multiply A(3, 5) x B(5, 2)

```
  My-code result: [
    [95, 110],
    [95, 110],
    [95, 110]
  ]
  Expected (strict): [
    [95, 110],
    [95, 110],
    [95, 110]
  ]
```

## *Test #15 (synthetic_1015)*

Java literal (verbatim):
```
Integer[][][] test = new Integer[][][] {{{1,2},{3}},{{4,5},{6,7},{8,9}}};
```

Actual Java Lists executed:
```
[
  [[1, 2], [3]],
  [[4, 5], [6, 7], [8, 9]]
]
```

System.out.println outputs (simulated mapping - none captured):
```
(none)
```

Step 1: Multiply A(2, 2) x B(3, 2)
```
  My-code result: [
    [16, 19],
    [12, 15]
  ]
  Expected (strict): Invalid/Discarded (Incompatible dims colsA=2 rowsB=3)
```

## *Test #16 (synthetic_1016)*

Java literal (verbatim):

```
Integer[][][] test = new Integer[][][] {{{1,2},{3,4}},{{5,6,7},{8,9,10}}};
```

Actual Java Lists executed:

```
[
  [[1, 2], [3, 4]],
  [[5, 6, 7], [8, 9, 10]]
]
```

System.out.println outputs (simulated mapping - none captured):

```
(none)
```

Step 1: Multiply A(2, 2) x B(2, 3)

```
  My-code result: [
    [21, 24, 27],
    [47, 54, 61]
  ]
  Expected (strict): [
    [21, 24, 27],
    [47, 54, 61]
  ]
```

## *Test #17 (synthetic_1017)*

Java literal (verbatim):
```
Integer[][][] test = new Integer[][][] {{{1,2}},{{}},{{3,4}}};
```

Actual Java Lists executed:
```
[
  [[1, 2]],
  [[]],
  [[3, 4]]
]
```

System.out.println outputs (simulated mapping - none captured):
```
(none)
```

Step 1: Multiply A(1, 2) x B(1, 0)
```
  My-code result: [
    []
  ]
  Expected (strict): Invalid/Discarded (Incompatible dims colsA=2 rowsB=1)
```

Step 2: Multiply A(1, 0) x B(1, 2)
```
  My-code result: [
    [0, 0]
  ]
  Expected (strict): Invalid/Discarded (Incompatible dims colsA=0 rowsB=1)
```

## *Test #18 (synthetic_1018)*

Java literal (verbatim):

```
Integer[][][] test = new Integer[][][] {{{1,2,3},{4,5}},{{6},{7},{8}}};
```

Actual Java Lists executed:

```
[
  [[1, 2, 3], [4, 5]],
  [[6], [7], [8]]
]
```

System.out.println outputs (simulated mapping - none captured):

```
(none)
```

Step 1: Multiply A(2, 3) x B(3, 1)

```
  My-code result: [
    [44],
    [59]
  ]
  Expected (strict): Invalid/Discarded (Invalid/Discarded due to missing A element)
```

## *Test #19 (synthetic_1019)*

Java literal (verbatim):

```
Integer[][][] test = new Integer[][][] {{{1,null,3,4},{5,6,null,8}},{{9,10},{11,12},{13,14},{15,16}}};
```

Actual Java Lists executed:

```
[
  [[1, None, 3, 4], [5, 6, None, 8]],
  [[9, 10], [11, 12], [13, 14], [15, 16]]
]
```

System.out.println outputs (simulated mapping - none captured):

```
(none)
```

Step 1: Multiply A(2, 4) x B(4, 2)

```
  My-code result: [
    [108, 116],
    [231, 250]
  ]
  Expected (strict): Invalid/Discarded (Invalid/Discarded due to missing A element)
```

## *Test #20 (synthetic_1020)*

Java literal (verbatim):

```
Integer[][][] test = new Integer[][][] {{{1,2}},{{3,4,5}},{{6,7}}};
```

Actual Java Lists executed:

```
[
  [[1, 2]],
  [[3, 4, 5]],
  [[6, 7]]
]
```

System.out.println outputs (simulated mapping - none captured):

```
(none)
```

Step 1: Multiply A(1, 2) x B(1, 3)

```
  My-code result: [
    [3, 4, 5]
  ]
  Expected (strict): Invalid/Discarded (Incompatible dims colsA=2 rowsB=1)
```

Step 2: Multiply A(1, 3) x B(1, 2)

```
  My-code result: [
    [18, 21]
  ]
  Expected (strict): Invalid/Discarded (Incompatible dims colsA=3 rowsB=1)
```

## *Test #21 (synthetic_1021)*

Java literal (verbatim):
```
Integer[][][] test = new Integer[][][] {{{5}},{{6}},{{7}}};
```

Actual Java Lists executed:
```
[
  [[5]],
  [[6]],
  [[7]]
]
```

System.out.println outputs (simulated mapping - none captured):
```
(none)
```

Step 1: Multiply A(1, 1) x B(1, 1)
```
  My-code result: [
    [30]
  ]
  Expected (strict): [
    [30]
  ]
```

Step 2: Multiply A(1, 1) x B(1, 1)
```
  My-code result: [
    [210]
  ]
  Expected (strict): [
    [210]
  ]
```

## *Test #22 (synthetic_1022)*

Java literal (verbatim):

```
Integer[][][] test = new Integer[][][] {{{0,0,0},{null,0}},{{0},{0},{0}}};
```

Actual Java Lists executed:

```
[
  [[0, 0, 0], [None, 0]],
  [[0], [0], [0]]
]
```

System.out.println outputs (simulated mapping - none captured):

```
(none)
```

Step 1: Multiply A(2, 3) x B(3, 1)

```
  My-code result: [
    [0],
    [0]
  ]
  Expected (strict): Invalid/Discarded (Invalid/Discarded due to missing A element)
```

## *Test #23 (synthetic_1023)*

Java literal (verbatim):

```
Integer[][][] test = new Integer[][][] {{{1,2,3,4}},{{1},{null},{3},{4}}};
```

Actual Java Lists executed:

```
[
  [[1, 2, 3, 4]],
  [[1], [None], [3], [4]]
]
```

System.out.println outputs (simulated mapping - none captured):

```
(none)
```

Step 1: Multiply A(1, 4) x B(4, 1)

```
  My-code result: [
    [26]
  ]
  Expected (strict): Invalid/Discarded (Invalid/Discarded due to missing B element)
```

## *Test #24 (synthetic_1024)*

Java literal (verbatim):
```
Integer[][][] test = new Integer[][][] {{{1,2},{3,4,5},{6}},{{7},{8},{9}}};
```

Actual Java Lists executed:
```
[
  [[1, 2], [3, 4, 5], [6]],
  [[7], [8], [9]]
]
```

System.out.println outputs (simulated mapping - none captured):
```
(none)
```

Step 1: Multiply A(3, 3) x B(3, 1)
```
  My-code result: [
    [23],
    [98],
    [42]
  ]
  Expected (strict): Invalid/Discarded (Invalid/Discarded due to missing A element)
```

## *Test #25 (synthetic_1025)*

Java literal (verbatim):

```
Integer[][][] test = new Integer[][][] {{{10,11,12},{13,14,15}},{{2,3},{4,5},{6,7}},{{1},{2}}};
```

Actual Java Lists executed:

```
[
  [[10, 11, 12], [13, 14, 15]],
  [[2, 3], [4, 5], [6, 7]],
  [[1], [2]]
]
```

System.out.println outputs (simulated mapping - none captured):

```
(none)
```

Step 1: Multiply A(2, 3) x B(3, 2)

```
  My-code result: [
    [136, 169],
    [172, 214]
  ]
  Expected (strict): [
    [136, 169],
    [172, 214]
  ]
```

Step 2: Multiply A(2, 2) x B(2, 1)

```
  My-code result: [
    [474],
    [600]
  ]
  Expected (strict): [
    [474],
    [600]
  ]
```

## *Test #26 (synthetic_1026)*

Java literal (verbatim):
```
Integer[][][] test = new Integer[][][] {{{1,2}},{{},{}},{{3,4}}};
```

Actual Java Lists executed:
```
[
  [[1, 2]],
  [[], []],
  [[3, 4]]
]
```

System.out.println outputs (simulated mapping - none captured):
```
(none)
```

Step 1: Multiply A(1, 2) x B(2, 0)
```
  My-code result: [
    []
  ]
  Expected (strict): [
    []
  ]
```

Step 2: Multiply A(1, 0) x B(1, 2)
```
  My-code result: [
    [0, 0]
  ]
  Expected (strict): Invalid/Discarded (Incompatible dims colsA=0 rowsB=1)
```

## *Test #27 (synthetic_1027)*

Java literal (verbatim):

```
Integer[][][] test = new Integer[][][] {{{1,2,3,4},{5,6,7,8},{9,10,11,12},{13,14,15,16}},{{16,15,14,13},{12,11,10,9},{8,7,6,5},{4,3,2,1}}};
```

Actual Java Lists executed:

```
[
  [[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12], [13, 14, 15, 16]],
  [[16, 15, 14, 13], [12, 11, 10, 9], [8, 7, 6, 5], [4, 3, 2, 1]]
]
```

System.out.println outputs (simulated mapping - none captured):

```
(none)
```

Step 1: Multiply A(4, 4) x B(4, 4)

```
  My-code result: [
    [80, 70, 60, 50],
    [240, 214, 188, 162],
    [400, 358, 316, 274],
    [560, 502, 444, 386]
  ]
  Expected (strict): [
    [80, 70, 60, 50],
    [240, 214, 188, 162],
    [400, 358, 316, 274],
    [560, 502, 444, 386]
  ]
```

## *Test #28 (synthetic_1028)*

Java literal (verbatim):
```
Integer[][][] test = new Integer[][][] {{{1,2,3},{},{4,5}},{{6},{7,8},{9,10,11}}};
```

Actual Java Lists executed:
```
[
  [[1, 2, 3], [], [4, 5]],
  [[6], [7, 8], [9, 10, 11]]
]
```

System.out.println outputs (simulated mapping - none captured):
```
(none)
```

Step 1: Multiply A(3, 3) x B(3, 3)
```
  My-code result: [
    [47, 46, 33],
    [null, null, null],
    [59, 40, null]
  ]
  Expected (strict): Invalid/Discarded (Invalid/Discarded due to missing A element)
```

## *Test #29 (synthetic_1029)*

Java literal (verbatim):

```
Integer[][][] test = new Integer[][][] {{{1,2,3},{4,5,6}},{{7},{8},{9}}};
```

Actual Java Lists executed:

```
[
  [[1, 2, 3], [4, 5, 6]],
  [[7], [8], [9]]
]
```

System.out.println outputs (simulated mapping - none captured):

```
(none)
```

Step 1: Multiply A(2, 3) x B(3, 1)

```
  My-code result: [
    [50],
    [122]
  ]
  Expected (strict): [
    [50],
    [122]
  ]
```

## *Test #30 (synthetic_1030)*

Java literal (verbatim):
```
Integer[][][] test = new Integer[][][] {{{3,1,8,null,0}},{{9}}};
```

Actual Java Lists executed:
```
[
  [[3, 1, 8, None, 0]],
  [[9]]
]
```

System.out.println outputs (simulated mapping - none captured):
```
(none)
```

Step 1: Multiply A(1, 5) x B(1, 1)
```
  My-code result: [
    [27]
  ]
  Expected (strict): Invalid/Discarded (Incompatible dims colsA=5 rowsB=1)
```

## Coverage Summary

AI-generated tests: 30

Loops visited across runs: 7

Loops not visited: 1, 2, 3, 4, 5, 6

## Closing Analysis

This synthetic suite stresses jagged chains, nulls, surplus elements, and mid-chain failures. Use a local Java harness to run these tests for exact outputs.