PU/NE-98-9

QUALIFICATION ASSESSMENT TEST PLAN FOR THE GENERAL INTERFACE IN THE COUPLED CODE

Douglas A. Barber, Thomas J. Downar

School of Nuclear Engineering
Purdue University
W. Lafayette, IN 47907-1290

Weidong Wang, Glen Mortensen

Scientech, Inc.
11140 Rockville Pike, Ste 500
Rockville, MD 20852

submitted to:

Division of Systems Technology
Office of Regulatory Research
Nuclear Regulatory Commission
Washington, D.C. 20555-001

Contract NRC-04-96-060, Task 5c
Performed Under Subcontract to SCIENTECH, Inc.

May 1998

The qualification assessment test plan for the General Interface in the Coupled Code will be consistent with the Software Requirement Specifications (SRS)[1] and performed as described below.

I. Test Plan Description

The test plan will be executed, as outlined below, using a five step approach to exercise each of the three functional units, the single error checking unit, and the process control of the General Interface. Note: for the purpose of this test plan, the input files are setup to perform two time-dependent calculations.

A. The first step will test the interface initialization using the following steps:

1. The data structure containing thermal-hydraulic control information will be received from the thermal-hydraulic mapping routine and sent to the neutronic mapping routine.
2. The data structure containing neutronic control information will be received from the neutronic mapping routine and sent to the thermal-hydraulic mapping routine.
3. Both floating point permutation matrices will be received from the thermal-hydraulic mapping routine and stored in the memory of the General Interface.
4. Both floating point permutation matrices will be received from the neutronic mapping routine and stored in the memory of the General Interface.

The determination to perform Step 3 or Step 4 will be based on the first value of the logical control buffers sent from the thermal-hydraulic and neutronic routines. Specifically, a value of TRUE in the seventh location of the thermal-hydraulic logical buffer will result in Step 3 being performed. Conversely, a value of TRUE in the seventh location of the neutronic logical buffer will result in Step 4 being performed.

B. The second step will test the thermal-hydraulics to neutronics mapping using the following steps:

1. A vector will be received from the thermal-hydraulics containing all the space-dependent thermal-hydraulic data.
2. The matrix-vector multiplication will be performed.
3. The resulting vector will be sent to the neutronics mapping routine.
4. The control data will be received from the thermal-hydraulic routine and transmitted to the neutronics.

C. The third step will test the neutronics to thermal-hydraulics mapping using the following steps:

1. A vector will be received from the neutronics containing all the space-dependent neutronics data.
2. The matrix-vector multiplication will be performed.
3. The resulting vector will be sent to the thermal-hydraulic mapping routine.
4. The control data will be received from the neutronics routine and transmitted to the thermal-hydraulics.

D. The fourth step will test the error checking for each of the previous three steps. In each case, the first test case described in Appendix A will be perturbed such that the parameter indicated triggers the appropriate message.

<u>Initialization</u>

1. The indication of which process (thermal-hydraulic or neutronic) is sending the permutation matrices is inconsistent [Fatal].
2. The matrix elements are outside the specified range [Fatal].
3. The weighting factors in the permutation matrix are inaccurate [Fatal].

<u>Thermal-Hydraulics to Neutronics Mapping</u>

1. Inconsistency between matrix and vector dimensions [Fatal].
2. Negative elements in vector [Fatal].

<u>Neutronics to Thermal-Hydraulics Mapping</u>

1. Inconsistency between matrix and vector dimensions [Fatal].
2. Negative elements in vector [Fatal].

E. The fifth step will test the termination of the General Interface based on the following conditions:

1. Normal termination resulting from logical information sent from both the thermal-hydraulics and neutronics code.
2. Termination based on a logical fault signal sent from either the thermal-hydraulics or the neutronics code.
3. Termination based on an error detected in the General Interface.

Note: Testing of the PVM error checking will not be performed for this QATP.

II. Test Cases

The test cases are chosen to provide coverage of the functionality of the General Interface, and are described in Appendix A of this report, along with the corresponding permutation matrices (Appendix B) and input vectors (Appendix C). In addition, the transfer of the control buffers through the General Interface will be tested using the data shown in Appendix E and F.

In order to exercise the error checking measures listed in Section I.D, permutations of the first test case, described in Appendices A through F, will be performed as follows:

Initialization

1a. The seventh element in the initial logical buffer for the neutronic process will be set to "F".
1b. The seventh element in the initial logical buffer for the thermal-hydraulic process will be set to "T".

2a. The first non-zero element in the thermal-hydraulic to neutronic matrix will be assigned a value of 2.0.
2b. The first non-zero element in the neutronic to thermal-hydraulic matrix will be assigned a value of 2.0.
2c. The first non-zero element in the thermal-hydraulic to neutronic matrix will be assigned a value of -1.0.
2d. The first non-zero element in the neutronic to thermal-hydraulic matrix will be assigned a value of -1.0.

3. The parametrics performed in 2a through 2d will trigger this error also.

Thermal-Hydraulics to Neutronics Mapping

1a. The dimension of the unpermuted vector of thermal-hydraulic and heat structure data will be decreased by one during the first time-dependent calculation.
1b. The dimension of the unpermuted vector of thermal-hydraulic and heat structure data will be decreased by one during the second time-dependent calculation. This test is performed because the General Interface only allocates space for the unpermuted vector once (at the first time step). Thus, if the dimension of the vector sent from the thermal-hydraulics process is different on subsequent time steps than the size of the previously allocated vector, an error should occur.

2. The first element in the unpermuted vector will be assigned a value of -1.0.

Neutronics to Thermal-Hydraulics Mapping

1a. The dimension of the unpermuted vector of neutronic data will be decreased by one during the first time-dependent calculation.

1b. The dimension of the unpermuted vector of neutronic data will be decreased by one during the second time-dependent calculation. The reason for this test is as described above.

2. The first element in both the unpermuted and permuted vectors will be assigned a value of -1.0.

Detection of faults originating in the thermal-hydraulics and neutronics codes will also be tested by assigning the first, second, and third logicals in the initial and time-dependent control buffers a value of "T". And finally, normal process termination will be tested based on a value of "T" for the sixth logical in the second time-dependent control buffer of both the thermal-hydraulics and neutronics code.

III. Test Plan Verification

The test plan will be verified using a script to compare the produced output to the expected output. The acceptance criteria are:

1. Every element of the input vectors and permutation matrices is correct compared with the data shown in the appendix.
2. Every element of the output vectors is correct compared with the data shown in the appendix. In addition, the output vectors computed in the General Interface are consistent with the vectors received in the thermal-hydraulics and neutronics codes.
3. The initial and time-dependent control buffers sent from the thermal-hydraulics code are the same as those received by the General Interface and neutronics codes.
4. The initial and time-dependent control buffers sent from the neutronics code are the same as those received by the General Interface and thermal-hydraulics codes.
5. All error measures are properly triggered.
6. Safe termination of the process is achieved.

IV. References

1. D. Barber and T. Downar, "Software Requirements Specification for the General Interface in the Coupled Code," Technical Report, PU/NE-98-8, Purdue University, (1998).

Appendix A: Description of Test Cases

The test matrix shown in Table 1, is used to test the functionality of the General Interface, where the number and type of independent solution variables (corresponding to the submatrices and subvectors) is described in Section IV.B. These cases will employ two different geometries for the thermal-hydraulic zones (i.e. cartesian and cylindrical) in order to exercise alternate mapping matrix structures.

Table 1: Test Case Matrix for General Interface

Case	Thermal-Hydraulic Geometry[a]	Thermal-Hydraulic Zones	Neutronic Radial Nodes	Neutronic Axial Nodes[b]	Heat Struct. Components
1[c]	Cartesian	4	4	1	4
2	Cartesian	1	16	1	1
3	Cartesian	16	4	1	16
4	Cylindrical	16	16	1	16
5	Cartesian	4	16	1	16
6	Cartesian	4	16	2	4
7[d]	Cylindrical	2	1	1	1

[a] Neutronic Geometry is always Cartesian
[b] Number of Thermal-Hydraulic and Heat Structure axial nodes is always 1
[c] Base test case corresponding to Figure: 1
[d] Test case for annulus problem corresponding to Figure: 2

<u>Case 1</u>

This is the base test case, as illustrated in Figure: 1, where the four heat structures are tied to four parallel thermal-hydraulic pipes. A one-to-one mapping exists between the thermal-hydraulic zones, heat structure components, and neutronic nodes.

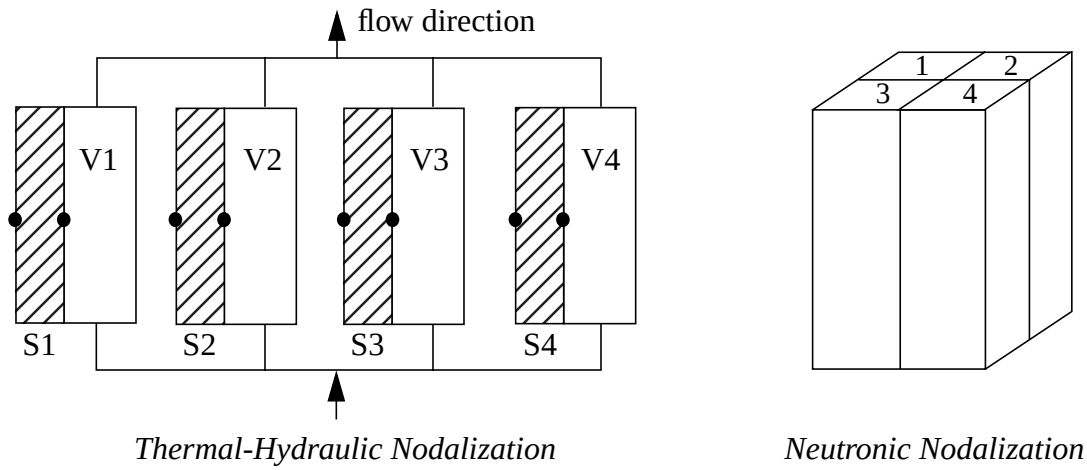

Figure 1: Problem Nodalization for Base Test Case

Case 2

This case represents a finer neutronic nodalization compared to that for the thermal-hydraulic zones and heat structure components. In addition, for the single heat structure, 17 radial mesh in the fuel pin is utilized to test the case of a well-described temperature profile. Note: for this case, the thermal-hydraulic code will construct and send the permutation matrices to the General Interface.

Case 3

This case represents a finer thermal-hydraulic and heat structure nodalization as compared to that for the neutronics. Here, the 16 parallel pipes are arranged in a cartesian 4x4 grid, and corresponds to a 2x2 grid for the neutronic nodes, such that each neutronic node contains four thermal-hydraulic zones and four heat structure components.

Case 4

This case tests a mapping from a cylindrical thermal-hydraulic nodalization to a cartesian neutronic nodalization. Here, 16 parallel thermal-hydraulic pipes, each with a single heat structure component, are arranged in a cylindrical grid, and mapped to 16 neutronic nodes arranged in a 4x4 grid, as demonstrated in Figure 5 of the SRS for the General Interface. This mapping is not one-to-one.

Case 5

This case tests the scenario where there are multiple heat structure components per thermal-hydraulic zone. A one-to-one correspondence exists between the heat structures and the neutronic

nodes, which are arranged in a 4x4 grid. The thermal-hydraulic zones are arranged in a 2x2 grid, and each thermal-hydraulic zone contains four heat structures. The mapping between thermal-hydraulic zones and neutronic nodes is demonstrated in Figure 4 of the SRS for the General Interface.

Case 6

This case is used for testing a 3-D mapping. Each thermal-hydraulic pipe, with a corresponding heat structure, is mapped to eight neutronic nodes (i.e. 2x2x2 layout).

Case 7

This annulus test case is set up to test the situation where a single heat structure is mapped to several thermal-hydraulic zones, and is illustrated in Figure: 2.

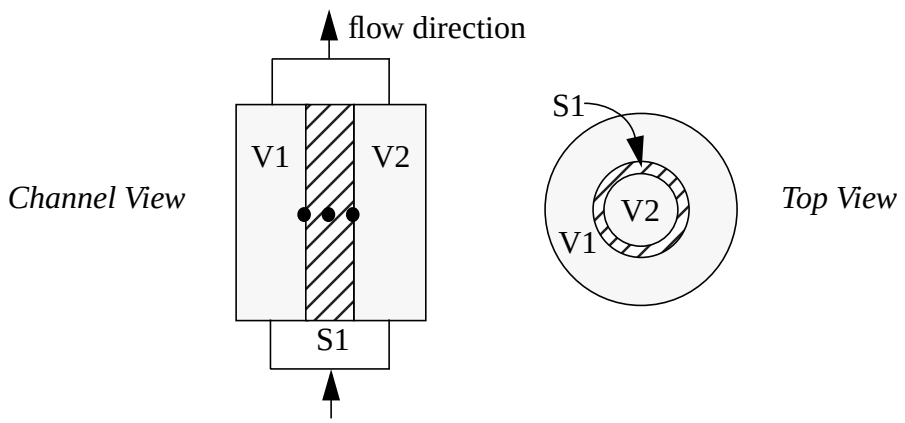

Figure 2: Thermal-Hydraulic Nodalization for Annulus Test Case

Appendix B: Permutation Submatrices in Coordinate Storage Format

The submatrices for the seven cases listed in Section II, are shown below. For each test case, the number and type of variables corresponding to each submatrix is as follows:

(A)	Thermal-Hydraulic to Neutronic:	5;	Mod. Temp., Liq. Dens., Vap. Dens., Void Fr., Boron Conc.
(B)	Heat Structure to Neutronic:	3;	Avg. Fuel Temp., Fuel Centerline Temp., Fuel Surface Temp.
(C)	Neutronic to Thermal-Hydraulic:	3;	Fission Power, FP Decay Power, Actinide Decay Power
(D)	Neutronic to Heat Structure:	3;	Fission Power, FP Decay Power, Actinide Decay Power

The entire permutation matrix consists of diagonal blocks corresponding to each submatrix. The order of the submatrices for each mapping direction is as follows:

Thermal-Hydraulic / Heat Structure to Neutronic:

$$\text{diag}\,[(A_1), (A_2), (A_3), (A_4), (A_5), (B_1), (B_2), (B_3)]$$

Neutronic to Thermal-Hydraulic / Heat Structure:

$$\text{diag}\,[(C_1), (C_2), (C_3), (D_1), (D_2), (D_3)]$$

It should be noted that the `ROW` and `COL` vectors shown for the test cases correspond to the local submatrices. The indices for the global permutation matrices required by the General Interface should be computed based on the described structure.

<u>Case 1:</u>

Thermal-Hydraulic to Neutronic

```
VAL = [1.0 1.0 1.0 1.0]
ROW = [ 1   2   3   4 ]
COL = [ 1   2   3   4 ]
```

Heat Structure to Neutronic

```
VAL = [1.0 1.0 1.0 1.0]
ROW = [ 1   2   3   4 ]
COL = [ 1   2   3   4 ]
```

Neutronic to Thermal-Hydraulic

```
VAL = [1.0 1.0 1.0 1.0]
ROW = [ 1   2   3   4 ]
COL = [ 1   2   3   4 ]
```

Neutronic to Heat Structure

```
VAL = [1.0 1.0 1.0 1.0]
ROW = [ 1   2   3   4 ]
COL = [ 1   2   3   4 ]
```

Case 2:

Thermal-Hydraulic to Neutronic

```
VAL = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0]
ROW = [ 1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16]
COL = [ 1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 ]
```

Heat Structure to Neutronic

```
VAL = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0]
ROW = [ 1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16]
COL = [ 1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 ]
```

Neutronic to Thermal-Hydraulic

```
VAL = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0]
ROW = [ 1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 ]
COL = [ 1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16]
```

Neutronic to Heat Structure

```
VAL = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0]
ROW = [ 1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 ]
COL = [ 1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16]
```

Case 3:

Thermal-Hydraulic to Neutronic

```
VAL = [.25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25]
ROW = [ 1   1   2   2   1   1   2   2   3   3   4   4   3   3   4   4 ]
COL = [ 1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16]
```

Heat Structure to Neutronic

```
VAL = [.25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25]
ROW = [ 1   1   2   2   1   1   2   2   3   3   4   4   3   3   4   4 ]
COL = [ 1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16]
```

Neutronic to Thermal-Hydraulic

```
VAL = [.25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25]
ROW = [ 1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16]
COL = [ 1   1   2   2   1   1   2   2   3   3   4   4   3   3   4   4 ]
```

Neutronic to Heat Structure

```
VAL = [.25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25]
ROW = [ 1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16]
COL = [ 1   1   2   2   1   1   2   2   3   3   4   4   3   3   4   4 ]
```

Case 4:

Thermal-Hydraulic to Neutronic

```
VAL = [.50 .50 .71 .29 .71 .29 .50 .50 .71 .29 .50 .50 .50 .50 .71 .29 ,
       .71 .29 .50 .50 .50 .50 .71 .29 .50 .50 .71 .29 .71 .29 .50 .50]
ROW = [ 1   1   2   2   3   3   4   4   5   5   6   6   7   7   8   8  ,
        9   9   10  10  11  11  12  12  13  13  14  14  15  15  16  16]
COL = [ 7   8   8   16  1   9   1   2   7   15  15  16  9   10  2   10 ,
        6   14  13  14  11  12  3   11  5   6   5   13  4   12  3   4 ]
```

Heat Structure to Neutronic

```
VAL = [.50 .50 .71 .29 .71 .29 .50 .50 .71 .29 .50 .50 .50 .50 .71 .29 ,
       .71 .29 .50 .50 .50 .50 .71 .29 .50 .50 .71 .29 .71 .29 .50 .50]
ROW = [ 1   1   2   2   3   3   4   4   5   5   6   6   7   7   8   8  ,
        9   9   10  10  11  11  12  12  13  13  14  14  15  15  16  16]
COL = [ 7   8   8   16  1   9   1   2   7   15  15  16  9   10  2   10 ,
        6   14  13  14  11  12  3   11  5   6   5   13  4   12  3   4 ]
```

Neutronic to Thermal-Hydraulic

```
VAL = [.50 .50 .71 .29 .71 .29 .50 .50 .71 .29 .50 .50 .50 .50 .71 .29 ,
       .71 .29 .50 .50 .50 .50 .71 .29 .50 .50 .71 .29 .71 .29 .50 .50]
ROW = [ 7   8   8   16  1   9   1   2   7   15  15  16  9   10  2   10 ,
        6   14  13  14  11  12  3   11  5   6   5   13  4   12  3   4 ]
COL = [ 1   1   2   2   3   3   4   4   5   5   6   6   7   7   8   8  ,
        9   9   10  10  11  11  12  12  13  13  14  14  15  15  16  16]
```

Neutronic to Heat Structure

```
VAL = [.50 .50 .71 .29 .71 .29 .50 .50 .71 .29 .50 .50 .50 .50 .71 .29 ,
       .71 .29 .50 .50 .50 .50 .71 .29 .50 .50 .71 .29 .71 .29 .50 .50]
ROW = [ 7   8   8   16  1   9   1   2   7   15  15  16  9   10  2   10 ,
        6   14  13  14  11  12  3   11  5   6   5   13  4   12  3   4 ]
COL = [ 1   1   2   2   3   3   4   4   5   5   6   6   7   7   8   8  ,
        9   9   10  10  11  11  12  12  13  13  14  14  15  15  16  16]
```

Case 5:

Thermal-Hydraulic to Neutronic

```
VAL = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0]
ROW = [ 1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16]
COL = [ 1   1   2   2   1   1   2   2   3   3   4   4   3   3   4   4 ]
```

Heat Structure to Neutronic

```
VAL = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0]
ROW = [ 1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16]
COL = [ 1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16]
```

Neutronic to Thermal-Hydraulic

```
VAL = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0]
ROW = [ 1   1   2   2   1   1   2   2   3   3   4   4   3   3   4   4 ]
COL = [ 1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16]
```

Neutronic to Heat Structure

```
VAL = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0]
ROW = [ 1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16]
COL = [ 1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16]
```

Case 6:

Thermal-Hydraulic to Neutronic

```
VAL = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,
       1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0]
ROW = [ 1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16 ,
        17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32]
COL = [ 1   1   2   2   1   1   2   2   3   3   4   4   3   3   4   4  ,
        1   1   2   2   1   1   2   2   3   3   4   4   3   3   4   4 ]
```

Heat Structure to Neutronic

```
VAL = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,
       1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0]
ROW = [ 1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16 ,
        17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32]
COL = [ 1   1   2   2   1   1   2   2   3   3   4   4   3   3   4   4  ,
        1   1   2   2   1   1   2   2   3   3   4   4   3   3   4   4 ]
```

Neutronic to Thermal-Hydraulic

```
VAL = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,
       1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0]
ROW = [ 1   1   2   2   1   1   2   2   3   3   4   4   3   3   4   4  ,
        1   1   2   2   1   1   2   2   3   3   4   4   3   3   4   4 ]
COL = [ 1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16 ,
        17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32]
```

Neutronic to Heat Structure

```
VAL = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,
       1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0]
ROW = [ 1   1   2   2   1   1   2   2   3   3   4   4   3   3   4   4  ,
        1   1   2   2   1   1   2   2   3   3   4   4   3   3   4   4 ]
COL = [ 1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16 ,
        17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32]
```

Case 7:

Thermal-Hydraulic to Neutronic

```
VAL = [ f  (1-f)] ;  f=0.23
ROW = [ 1    1  ]
COL = [ 1    2  ]
```

Heat Structure to Neutronic

```
VAL = [ 1 ]
ROW = [ 1 ]
COL = [ 1 ]
```

Neutronic to Thermal-Hydraulic

```
VAL = [ f  (1-f)] ;  f=0.23
ROW = [ 1    2  ]
COL = [ 1    1  ]
```

Neutronic to Heat Structure

```
VAL = [ 1 ]
ROW = [ 1 ]
COL = [ 1 ]
```

Appendix C: Input Vectors for Permutation

The number and types of variables (subvectors) used for each test case was shown in the description of the permutation matrices. The order of the variables in the entire input vector is consistent with that for the matrix. The elements of the input vectors for each mapping direction are shown below for each test case, where `vecth` is the unpermuted vector of thermal-hydraulic and heat structure data and `vecn` is the unpermuted vector of neutronic data. An example of the order of subvectors is shown in Case 1, and applies to Cases 2 - 7. It should be noted that two time-dependent calculations are being performed, and the input vectors shown below are the same during both "time steps".

<u>Case 1:</u>

Thermal-Hydraulic / Heat Structure to Neutronic
(5 T/H vars, 4 T/H zones, 3 HS vars, 4 HS components: 32 elements)

```
         |   Mod. Temp.  |   Liq. Dens.  |   Vap. Dens.  |    Void Fr.   |
vecth^T = [1.0 2.0 3.0 4.0 3.0 3.0 3.0 3.0 1.0 1.0 1.0 1.0 0.5 0.5 1.0 0.0 ,
         |  Boron Conc.  |Avg. Fuel Temp.|  Center Temp. | Surface Temp. |
          4.0 2.0 4.0 3.0 5.0 5.0 5.0 5.0 6.0 6.0 6.0 6.0 4.0 4.0 4.0 4.0]
```

Neutronic to Thermal-Hydraulic / Heat Structure
(3 T/H vars, 4 Neut. nodes, 3 HS vars, 4 Neut. nodes: 24 elements)

```
         |            <-- Mapped to Coolant -->          |
         | Fission Power |   FP Power    |Actinide Power |
 vecn^T = [3.0 3.0 3.0 3.0 4.0 4.0 4.0 4.0 5.0 5.0 5.0 5.0 ,
         |            <--  Mapped to Fuel   -->          |
         | Fission Power |   FP Power    |Actinide Power |
          6.0 6.0 6.0 6.0 7.0 7.0 7.0 7.0 8.0 8.0 8.0 8.0]
```

<u>Case 2:</u>

Thermal-Hydraulic / Heat Structure to Neutronic
(5 T/H vars, 1 T/H zone, 3 HS vars, 1 HS component: 8 elements)

```
vecth^T = [1.0 3.0 1.0 0.5 4.0 5.0 6.0 4.0]
```

Neutronic to Thermal-Hydraulic / Heat Structure
(3 T/H vars, 16 Neut. nodes, 3 HS vars, 16 Neut. nodes: 96 elements)

```
 vecn^T = [3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 ,
           4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 ,
           5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,
           6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
           7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 ,
           8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0]
```

Case 3:

Thermal-Hydraulic / Heat Structure to Neutronic
(5 T/H vars, 16 T/H zones, 3 HS vars, 16 HS components: 128 elements)

```
vecth^T = [1.0 2.0 3.0 4.0 1.0 2.0 3.0 4.0 1.0 2.0 3.0 4.0 1.0 2.0 3.0 4.0 ,
           3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 ,
           1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,
           0.5 0.5 1.0 0.0 0.5 0.5 1.0 0.0 0.5 0.5 1.0 0.0 0.5 0.5 1.0 0.0 ,
           4.0 2.0 4.0 3.0 4.0 2.0 4.0 3.0 4.0 2.0 4.0 3.0 4.0 2.0 4.0 3.0 ,
           5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,
           6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
           4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0]
```

Neutronic to Thermal-Hydraulic / Heat Structure
(3 T/H vars, 4 Neut. nodes, 3 HS vars, 4 Neut. nodes: 24 elements)

```
 vecn^T = [3.0 3.0 3.0 3.0 4.0 4.0 4.0 4.0 5.0 5.0 5.0 5.0 6.0 6.0 6.0 6.0 ,
           7.0 7.0 7.0 7.0 8.0 8.0 8.0 8.0]
```

Case 4:

Thermal-Hydraulic / Heat Structure to Neutronic
(5 T/H vars, 16 T/H zones, 3 HS vars, 16 HS components: 128 elements)

```
vecth^T = [1.0 2.0 3.0 4.0 1.0 2.0 3.0 4.0 1.0 2.0 3.0 4.0 1.0 2.0 3.0 4.0 ,
           3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 ,
           1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,
           0.5 0.5 1.0 0.0 0.5 0.5 1.0 0.0 0.5 0.5 1.0 0.0 0.5 0.5 1.0 0.0 ,
           4.0 2.0 4.0 3.0 4.0 2.0 4.0 3.0 4.0 2.0 4.0 3.0 4.0 2.0 4.0 3.0 ,
           5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,
           6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
           4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0]
```

Neutronic to Thermal-Hydraulic / Heat Structure
(3 T/H vars, 16 Neut. nodes, 3 HS vars, 16 Neut. nodes: 96 elements)

```
 vecn^T = [3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 ,
           4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 ,
           5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,
           6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
           7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 ,
           8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0]
```

Case 5:

Thermal-Hydraulic / Heat Structure to Neutronic
(5 T/H vars, 4 T/H zones, 3 HS vars, 16 HS components: 68 elements)

```
vecth^T = [1.0 2.0 3.0 4.0 3.0 3.0 3.0 3.0 1.0 1.0 1.0 1.0 0.5 0.5 1.0 0.0 ,
           4.0 2.0 4.0 3.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,
           5.0 5.0 5.0 5.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
           6.0 6.0 6.0 6.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 ,
           4.0 4.0 4.0 4.0]
```

Neutronic to Thermal-Hydraulic / Heat Structure
(3 T/H vars, 16 Neut. nodes, 3 HS vars, 16 Neut. nodes: 96 elements)

```
 vecn^T = [3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 ,
           4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 ,
           5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,
           6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
           7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 ,
           8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0]
```

Case 6:

Thermal-Hydraulic / Heat Structure to Neutronic
(5 T/H vars, 4 T/H zones, 3 HS vars, 4 HS components: 32 elements)

```
vecth^T = [1.0 2.0 3.0 4.0 3.0 3.0 3.0 3.0 1.0 1.0 1.0 1.0 0.5 0.5 1.0 0.0 ,
           4.0 2.0 4.0 3.0 5.0 5.0 5.0 5.0 6.0 6.0 6.0 6.0 4.0 4.0 4.0 4.0]
```

Neutronic to Thermal-Hydraulic / Heat Structure
(3 T/H vars, 32 Neut. nodes, 3 HS vars, 32 Neut. nodes: 192 elements)

```
 vecn^T = [3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 ,
           3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 ,
           4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 ,
           4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 ,
           5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,
           5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,
           6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
           6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
           7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 ,
           7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 ,
           8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 ,
           8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0]
```

Case 7:

Thermal-Hydraulic / Heat Structure to Neutronic
(5 T/H vars, 2 T/H zones, 3 HS vars, 1 HS component: 13 elements)

$\mathtt{vecth}^T$ = [1.0 2.0 3.0 3.0 1.0 1.0 0.5 1.0 4.0 2.0 5.0 6.0 4.0]

Neutronic to Thermal-Hydraulic / Heat Structure
(3 T/H vars, 1 Neut. node, 3 HS vars, 1 Neut. node: 6 elements)

$\mathtt{vecn}^T$ = [3.0 4.0 5.0 6.0 7.0 8.0]

Appendix D: Output Vectors After Permutation

The output vector resulting from the permutation is shown below for each test case, where `vecthp` is the permuted vector of thermal-hydraulic and heat structure data and `vecnp` is the permuted vector of neutronic data. These vectors were obtained by performing the matrix-vector multiply electronically with MATLAB. Visual verification of the input vectors, input matrices, and output vectors was performed.

Case 1:

Thermal-Hydraulic / Heat Structure to Neutronic
(5 T/H vars, 4 Neut. nodes, 3 HS vars, 4 Neut. nodes: 32 elements)

```
vecthp^T = [1.0 2.0 3.0 4.0 3.0 3.0 3.0 3.0 1.0 1.0 1.0 1.0 0.5 0.5 1.0 0.0 ,
            4.0 2.0 4.0 3.0 5.0 5.0 5.0 5.0 6.0 6.0 6.0 6.0 4.0 4.0 4.0 4.0]
```

Neutronic to Thermal-Hydraulic / Heat Structure
(3 T/H vars, 4 T/H zones, 3 HS vars, 4 HS components: 24 elements)

```
 vecnp^T = [3.0 3.0 3.0 3.0 4.0 4.0 4.0 4.0 5.0 5.0 5.0 5.0 6.0 6.0 6.0 6.0 ,
            7.0 7.0 7.0 7.0 8.0 8.0 8.0 8.0]
```

Case 2:

Thermal-Hydraulic / Heat Structure to Neutronic
(5 T/H vars, 16 Neut. nodes, 3 HS vars, 16 Neut. nodes: 128 elements)

```
vecthp^T = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,
            3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 ,
            1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,
            0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ,
            4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 ,
            5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,
            6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
            4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0]
```

Neutronic to Thermal-Hydraulic / Heat Structure
(3 T/H vars, 1 T/H zone, 3 HS vars, 1 HS component: 6 elements)

```
 vecnp^T = [48.0 64.0 80.0 96.0 112.0 128.0]
```

Case 3:

Thermal-Hydraulic / Heat Structure to Neutronic
(5 T/H vars, 4 Neut. nodes, 3 HS vars, 4 Neut. nodes: 32 elements)

```
vecthp^T = [1.5 3.5 1.5 3.5 3.0 3.0 3.0 3.0 1.0 1.0 1.0 1.0 0.5 0.5 0.5 0.5 ,
```

```
           3.0 3.5 3.0 3.5 5.0 5.0 5.0 5.0 6.0 6.0 6.0 6.0 4.0 4.0 4.0 4.0]
```

Neutronic to Thermal-Hydraulic / Heat Structure
(3 T/H vars, 16 T/H zones, 3 HS vars, 16 HS components: 96 elements)

```
 vecnp^T = [0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 ,
            0.75 0.75 0.75 0.75 1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  ,
            1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.25 1.25 1.25 1.25 ,
            1.25 1.25 1.25 1.25 1.25 1.25 1.25 1.25 1.25 1.25 1.25 1.25 ,
            1.5  1.5  1.5  1.5  1.5  1.5  1.5  1.5  1.5  1.5  1.5  1.5  ,
            1.5  1.5  1.5  1.5  1.75 1.75 1.75 1.75 1.75 1.75 1.75 1.75 ,
            1.75 1.75 1.75 1.75 1.75 1.75 1.75 1.75 2.0  2.0  2.0  2.0  ,
            2.0  2.0  2.0  2.0  2.0  2.0  2.0  2.0  2.0  2.0  2.0  2.0]
```

Case 4:

Thermal-Hydraulic / Heat Structure to Neutronic
(5 T/H vars, 16 Neut. nodes, 3 HS vars, 16 Neut. nodes: 128 elements)

```
vecthp^T = [3.5 4.0 1.0 1.5 3.0 3.5 1.5 2.0 2.0 1.5 3.5 3.0 1.5 1.0 4.0 3.5 ,
            3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 ,
            1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,
            0.5 0.0 0.5 0.5 1.0 0.5 0.5 0.5 0.5 0.5 0.5 1.0 0.5 0.5 0.0 0.5 ,
            3.5 3.0 4.0 3.0 4.0 3.5 3.0 2.0 2.0 3.0 3.5 4.0 3.0 4.0 3.0 3.5 ,
            5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,
            6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
            4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0]
```

Neutronic to Thermal-Hydraulic / Heat Structure
(3 T/H vars, 16 T/H zones, 3 HS vars, 16 HS components: 96 elements)

```
 vecnp^T = [3.63 3.63 3.63 3.63 3.63 3.63 3.63 3.63 2.37 2.37 2.37 2.37 ,
            2.37 2.37 2.37 2.37 4.84 4.84 4.84 4.84 4.84 4.84 4.84 4.84 ,
            3.16 3.16 3.16 3.16 3.16 3.16 3.16 3.16 6.05 6.05 6.05 6.05 ,
            6.05 6.05 6.05 6.05 3.95 3.95 3.95 3.95 3.95 3.95 3.95 3.95 ,
            7.26 7.26 7.26 7.26 7.26 7.26 7.26 7.26 4.74 4.74 4.74 4.74 ,
            4.74 4.74 4.74 4.74 8.47 8.47 8.47 8.47 8.47 8.47 8.47 8.47 ,
            5.53 5.53 5.53 5.53 5.53 5.53 5.53 5.53 9.68 9.68 9.68 9.68 ,
            9.68 9.68 9.68 9.68 6.32 6.32 6.32 6.32 6.32 6.32 6.32 6.32]
```

Case 5:

Thermal-Hydraulic / Heat Structure to Neutronic
(5 T/H vars, 16 Neut. nodes, 3 HS vars, 16 Neut. nodes: 128 elements)

```
vecthp^T = [1.0 1.0 2.0 2.0 1.0 1.0 2.0 2.0 3.0 3.0 4.0 4.0 3.0 3.0 4.0 4.0 ,
            3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 ,
            1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,
            0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 1.0 1.0 0.0 0.0 1.0 1.0 0.0 0.0 ,
            4.0 4.0 2.0 2.0 4.0 4.0 2.0 2.0 4.0 4.0 3.0 3.0 4.0 4.0 3.0 3.0 ,
            5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,
```

```
          6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
          4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0]
```

Neutronic to Thermal-Hydraulic / Heat Structure
(3 T/H vars, 4 T/H zones, 3 HS vars, 16 HS components: 60 elements)

```
 vecnp^T = [12.0 12.0 12.0 12.0 16.0 16.0 16.0 16.0 20.0 20.0 20.0 20.0 ,
            6.0  6.0  6.0  6.0  6.0  6.0  6.0  6.0  6.0  6.0  6.0  6.0 ,
            6.0  6.0  6.0  6.0  7.0  7.0  7.0  7.0  7.0  7.0  7.0  7.0 ,
            7.0  7.0  7.0  7.0  7.0  7.0  7.0  7.0  8.0  8.0  8.0  8.0 ,
            8.0  8.0  8.0  8.0  8.0  8.0  8.0  8.0  8.0  8.0  8.0  8.0]
```

Case 6:

Thermal-Hydraulic / Heat Structure to Neutronic
(5 T/H vars, 32 Neut. nodes, 3 HS vars, 32 Neut. nodes: 256 elements)

```
vecthp^T = [1.0 1.0 2.0 2.0 1.0 1.0 2.0 2.0 3.0 3.0 4.0 4.0 3.0 3.0 4.0 4.0 ,
            1.0 1.0 2.0 2.0 1.0 1.0 2.0 2.0 3.0 3.0 4.0 4.0 3.0 3.0 4.0 4.0 ,
            3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 ,
            3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 ,
            1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,
            1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,
            0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 1.0 1.0 0.0 0.0 1.0 1.0 0.0 0.0 ,
            0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 1.0 1.0 0.0 0.0 1.0 1.0 0.0 0.0 ,
            4.0 4.0 2.0 2.0 4.0 4.0 2.0 2.0 4.0 4.0 3.0 3.0 4.0 4.0 3.0 3.0 ,
            4.0 4.0 2.0 2.0 4.0 4.0 2.0 2.0 4.0 4.0 3.0 3.0 4.0 4.0 3.0 3.0 ,
            5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,
            5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,
            6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
            6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
            4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 ,
            4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0]
```

Neutronic to Thermal-Hydraulic / Heat Structure
(3 T/H vars, 4 T/H zones, 3 HS vars, 4 HS components: 24 elements)

```
 vecnp^T = [24.0 24.0 24.0 24.0 32.0 32.0 32.0 32.0 40.0 40.0 40.0 40.0 ,
            48.0 48.0 48.0 48.0 56.0 56.0 56.0 56.0 64.0 64.0 64.0 64.0]
```

Case 7:

Thermal-Hydraulic / Heat Structure to Neutronic
(5 T/H vars, 1 Neut. node, 3 HS vars, 1 Neut. node: 8 elements)

```
vecthp^T = [1.77 3.0  1.0  0.885 2.46 5.0  6.0  4.0]
```

Neutronic to Thermal-Hydraulic / Heat Structure
(3 T/H vars, 2 T/H zones, 3 HS vars, 1 HS components: 9 elements)

```
 vecnp^T = [0.69 2.31 0.92 3.08 1.15 3.85 6.0  7.0  8.0]
```

Appendix E: Initial Control Buffers

The data used to test the transfer of the initial control buffers is show below for each test case. All of the data shown represent 'dummy' values except the following:

1. The first element in the logical buffer is used for communicating a calculation error in the respective code (neutronic or thermal-hydraulic).
2. The second element in the logical buffer is used for communicating a data error in the code-specific data map routine.
3. The third element in the logical buffer is used for communicating a PVM error in the code-specific data map routine
4. The fourth element in the logical buffer is used by the General Interface to communicate a data error to the respective code.
5. The fifth element in the logical buffer is used by the General Interface to communicate a PVM error to the respective code.
6. The sixth element in the logical buffer is used to communicate a normal calculation termination.
7. The seventh element in the logical buffer tells the General Interface which code is sending the permutation matrices (e.g. the code sending the matrices will have a value of "T" in this location).

Case 1:

	Neutronics Code	Thermal-Hydraulics Code
char*6	["test01", "test02"]	["test03", "test04"]
logical	[F, F, F, F, F, F, T]	[F, F, F, F, F, F, F]
integer*2	[0, 2, 1]	[1, 0, 2]
integer*4	[888]	[999]
real*4	[1.0, 2.0, 3.0]	[4.0, 5.0, 6.0]
real*8	[10.0, 20.0, 30.0]	[40.0, 50.0, 60.0]

Case 2:

	Neutronics Code	Thermal-Hydraulics Code
char*6	["test01", "test02"]	["test03", "test04"]
logical	[F, F, F, F, F, F, F]	[F, F, F, F, F, F, T]
integer*2	[0, 2, 1]	[1, 0, 2]
integer*4	[888]	[999]
real*4	[1.0, 2.0, 3.0]	[4.0, 5.0, 6.0]
real*8	[10.0, 20.0, 30.0]	[40.0, 50.0, 60.0]

Case 3:

	Neutronics Code	Thermal-Hydraulics Code
char*6	[“test01”, ”test02”]	[“test03”, “test04”]
logical	[F, F, F, F, F, F, T]	[F, F, F, F, F, F, F]
integer*2	[0, 2, 1]	[999]
integer*4	[888]	[16, 16]
real*4	[1.0, 2.0, 3.0]	[4.0, 5.0, 6.0]
real*8	[10.0, 20.0, 30.0]	[40.0, 50.0, 60.0]

Case 4:

	Neutronics Code	Thermal-Hydraulics Code
char*6	[“test01”, ”test02”]	[“test03”, “test04”]
logical	[F, F, F, F, F, F, T]	[F, F, F, F, F, F, F]
integer*2	[0, 2, 1]	[1, 0, 2]
integer*4	[888]	[999]
real*4	[1.0, 2.0, 3.0]	[4.0, 5.0, 6.0]
real*8	[10.0, 20.0, 30.0]	[40.0, 50.0, 60.0]

Case 5:

	Neutronics Code	Thermal-Hydraulics Code
char*6	[“test01”, ”test02”]	[“test03”, “test04”]
logical	[F, F, F, F, F, F, T]	[F, F, F, F, F, F, F]
integer*2	[0, 2, 1]	[1, 0, 2]
integer*4	[888]	[999]
real*4	[1.0, 2.0, 3.0]	[4.0, 5.0, 6.0]
real*8	[10.0, 20.0, 30.0]	[40.0, 50.0, 60.0]

Case 6:

	Neutronics Code	Thermal-Hydraulics Code
char*6	[“test01”, ”test02”]	[“test03”, “test04”]
logical	[F, F, F, F, F, F, T]	[F, F, F, F, F, F, F]
integer*2	[0, 2, 1]	[1, 0, 2]
integer*4	[888]	[999]
real*4	[1.0, 2.0, 3.0]	[4.0, 5.0, 6.0]
real*8	[10.0, 20.0, 30.0]	[40.0, 50.0, 60.0]

Case 7:

	Neutronics Code	Thermal-Hydraulics Code
`char*6`	`[“test01”, ”test02”]`	`[“test03”, “test04”]`
`logical`	`[F, F, F, F, F, F, T]`	`[F, F, F, F, F, F, F]`
`integer*2`	`[0, 2, 1]`	`[1, 0, 2]`
`integer*4`	`[888]`	`[999]`
`real*4`	`[1.0, 2.0, 3.0]`	`[4.0, 5.0, 6.0]`
`real*8`	`[10.0, 20.0, 30.0]`	`[40.0, 50.0, 60.0]`

Appendix F: Time-Dependent Control Buffers

The data used to test the transfer of the time-dependent control buffers is show below for each test case. Two time-dependent calculations are performed, and all of the data in the control buffers remains the same during both "time steps" except for the sixth logical in both the thermal-hydraulic and neutronic control buffers, which is used to indicate a normal calculation termination. All of the data shown in the buffers represent 'dummy' values except the following:

1. The first element in the logical buffer is used for communicating a calculation error in the respective code (neutronic or thermal-hydraulic).
2. The second element in the logical buffer is used for communicating a data error in the code-specific data map routine.
3. The third element in the logical buffer is used for communicating a PVM error in the code-specific data map routine
4. The fourth element in the logical buffer is used by the General Interface to communicate a data error to the respective code.
5. The fifth element in the logical buffer is used by the General Interface to communicate a PVM error to the respective code.
6. The sixth element in the logical buffer is used to communicate a normal calculation termination.

Case 1:

	Neutronics Code	Thermal-Hydraulics Code
char*6	["test05", "test06"]	["test07", "test08"]
logical (1st)	[F, F, F, F, F, F]	[F, F, F, F, F, F]
logical (2nd)	[F, F, F, F, F, T]	[F, F, F, F, F, T]
integer*2	[3, 4, 5]	[6, 7, 8]
integer*4	[888]	[999]
real*4	[7.0, 8.0, 9.0]	[10.0, 11.0, 12.0]
real*8	[70.0, 80.0, 90.0]	[100.0, 110.0, 120.0]

Case 2:

	Neutronics Code	Thermal-Hydraulics Code
char*6	["test05", "test06"]	["test07", "test08"]
logical (1st)	[F, F, F, F, F, F]	[F, F, F, F, F, F]
logical (2nd)	[F, F, F, F, F, T]	[F, F, F, F, F, T]
integer*2	[3, 4, 5]	[6, 7, 8]
integer*4	[888]	[999]
real*4	[7.0, 8.0, 9.0]	[10.0, 11.0, 12.0]
real*8	[70.0, 80.0, 90.0]	[100.0, 110.0, 120.0]

Case 3:

	Neutronics Code	Thermal-Hydraulics Code
char*6	[“test05”, ”test06”]	[“test07”, “test08”]
logical (1st)	[F, F, F, F, F, F]	[F, F, F, F, F, F]
logical (2nd)	[F, F, F, F, F, T]	[F, F, F, F, F, T]
integer*2	[3, 4, 5]	[6, 7, 8]
integer*4	[888]	[999]
real*4	[7.0, 8.0, 9.0]	[10.0, 11.0, 12.0]
real*8	[70.0, 80.0, 90.0]	[100.0, 110.0, 120.0]

Case 4:

	Neutronics Code	Thermal-Hydraulics Code
char*6	[“test05”, ”test06”]	[“test07”, “test08”]
logical (1st)	[F, F, F, F, F, F]	[F, F, F, F, F, F]
logical (2nd)	[F, F, F, F, F, T]	[F, F, F, F, F, T]
integer*2	[3, 4, 5]	[6, 7, 8]
integer*4	[888]	[999]
real*4	[7.0, 8.0, 9.0]	[10.0, 11.0, 12.0]
real*8	[70.0, 80.0, 90.0]	[100.0, 110.0, 120.0]

Case 5:

	Neutronics Code	Thermal-Hydraulics Code
char*6	[“test05”, ”test06”]	[“test07”, “test08”]
logical (1st)	[F, F, F, F, F, F]	[F, F, F, F, F, F]
logical (2nd)	[F, F, F, F, F, T]	[F, F, F, F, F, T]
integer*2	[3, 4, 5]	[6, 7, 8]
integer*4	[888]	[999]
real*4	[7.0, 8.0, 9.0]	[10.0, 11.0, 12.0]
real*8	[70.0, 80.0, 90.0]	[100.0, 110.0, 120.0]

Case 6:

	Neutronics Code	Thermal-Hydraulics Code
char*6	[“test05”, ”test06”]	[“test07”, “test08”]
logical (1st)	[F, F, F, F, F, F]	[F, F, F, F, F, F]
logical (2nd)	[F, F, F, F, F, T]	[F, F, F, F, F, T]
integer*2	[3, 4, 5]	[6, 7, 8]
integer*4	[888]	[999]
real*4	[7.0, 8.0, 9.0]	[10.0, 11.0, 12.0]
real*8	[70.0, 80.0, 90.0]	[100.0, 110.0, 120.0]

Case 7:

	Neutronics Code	Thermal-Hydraulics Code
`char*6`	`[“test05”, ”test06”]`	`[“test07”, “test08”]`
`logical (1st)`	`[F, F, F, F, F, F]`	`[F, F, F, F, F, F]`
`logical (2nd)`	`[F, F, F, F, F, T]`	`[F, F, F, F, F, T]`
`integer*2`	`[3, 4, 5]`	`[6, 7, 8]`
`integer*4`	`[888]`	`[999]`
`real*4`	`[7.0, 8.0, 9.0]`	`[10.0, 11.0, 12.0]`
`real*8`	`[70.0, 80.0, 90.0]`	`[100.0, 110.0, 120.0]`