A solution mapping algorithm in EUROPLEXUS

Casadei, F.
Larcher, M.
Valsamos, G.
Aune, V.

2018
A solution mapping algorithm in EUROPLEXUS

F. Casadei¹, M. Larcher², G. Valsamos², and V. Aune³

¹Active Senior, European Commission, Joint Research Centre, Directorate for Space Security and Migration, Safety and security of buildings Unit
²European Commission, Joint Research Centre, Directorate for Space Security and Migration, Safety and security of buildings Unit
³Structural Impact Laboratory (SIMLab), Centre for Research-based Innovation (CRI) and Department of Structural Engineering, Norwegian University of Science and Technology, Rich. Birkelands vei 1A, NO-7491 Trondheim, Norway

May 20, 2018

Contents

1 Introduction 3

2 Mapping commands 4
  2.1 The ECRI FICH MAPP directive ........................................ 4
  2.2 The INIT MAPP directive .................................................. 4
  2.3 Limitations ................................................................. 5

3 Implementation notes 5
  3.1 Data structure .............................................................. 5
  3.2 Fast search of the mapping correspondence .......................... 6

4 Numerical examples 7
  4.1 Simple 2D tests .......................................................... 7
    4.1.1 Case MAPP00 .......................................................... 7
    4.1.2 Case MAPP01 .......................................................... 8
    4.1.3 Case MAPP02 .......................................................... 8
    4.1.4 Cases MAPP03 and MAPP04 ......................................... 9
    4.1.5 Cases MAPP05 and MAPP06 ......................................... 10
  4.2 Simple 3D tests .......................................................... 11
    4.2.1 Case MAPP09 .......................................................... 11
    4.2.2 Cases MAPP07 and MAPP08 ......................................... 12
  4.3 Simple 1D tests .......................................................... 13
    4.3.1 Case MAPP10 .......................................................... 13
    4.3.2 Cases MAPP11 and MAPP12 ......................................... 13
    4.3.3 Cases MAPP17 and MAPP18 ......................................... 13
  4.4 Simple combined 1D/3D tests .......................................... 14
    4.4.1 Case MAPP20 .......................................................... 14
    4.4.2 Case MAPP21 .......................................................... 16
    4.4.3 Case MAPP22 .......................................................... 16
  4.5 3D tests ................................................................. 16
    4.5.1 Case MAPP30 .......................................................... 17
    4.5.2 Case MAPP31 .......................................................... 18
    4.5.3 Case MAPP32 .......................................................... 18
  4.6 NTNU shock tube tests ................................................ 19
    4.6.1 Case VEGA51 ........................................................ 19
4.6.2 Case VEGA52 .......................................................... 20
4.6.3 Case VEGA61 .......................................................... 21
4.6.4 Case VEGA62 .......................................................... 23
4.6.5 Comparison of results .............................................. 24
4.6.6 Case VEGA53 .......................................................... 25
4.6.7 Case VEGA63 .......................................................... 26
4.6.8 Comparison of first membrane failure patterns ............... 27

5 Practical notes 35

References 35

Appendix I — Input files 36

List of input files 71

List of Tables

1 Simple 2D numerical simulations. ..................................... 7
2 Simple 3D numerical simulations. ..................................... 11
3 Simple 1D numerical simulations. ..................................... 13
4 Simple 1D/3D numerical simulations. ................................. 15
5 More complex 3D numerical simulations. ............................ 17
6 NTNU shock tube simulations. ........................................... 19

List of Figures

1 Geometry of test case MAPP00. ........................................ 7
2 Results of test case MAPP00. ........................................... 8
3 Comparison of results of tests MAPP01 and MAPP02 for \( t \geq 50 \) ms. ......................................................... 9
4 Comparison of results of tests MAPP03 and MAPP04 for \( t \geq 50 \) ms. ......................................................... 10
5 Models of tests MAPP05 and MAPP06. ................................. 11
6 Results of test case MAPP09. ........................................... 12
7 Comparison of results of test case MAPP00 (first order) and MAPP09 (second order). 12
8 Comparison of results of tests MAPP07 and MAPP08 for \( t \geq 50 \) ms. ......................................................... 13
9 Geometry of test case MAPP10. ........................................ 14
10 Results of test case MAPP10. ........................................... 14
11 Comparison of results of tests MAPP11 and MAPP12 for \( t \geq 50 \) ms. ......................................................... 15
12 Comparison of results of tests MAPP17 and MAPP18 for \( t \geq 50 \) ms. ......................................................... 15
13 Mixed 1D-3D model used in test case MAPP20. ....................... 15
14 Results of test case MAPP20. ........................................... 16
15 Comparison of results of tests MAPP21 and MAPP22 at \( t = 20 \) ms. ......................................................... 17
16 Geometry for test case MAPP30. ........................................ 17
17 Fluid pressures in tests MAPP30. ....................................... 18
18 Geometry for test case MAPP31. ........................................ 18
19 Comparison of results of tests MAPP30 and MAPP32. ............... 19
20 Results of test VEGA51. .................................................. 20
21 Results of test VEGA52. .................................................. 21
22 Comparison of results of tests VEGA51 and VEGA52. ............... 21
23 Comparison of results of tests VEGA51 and VEGA52 (zoom). ....... 22
24 Results of test VEGA62. .................................................. 23
25 Comparison of results of tests VEGA51, VEGA52 and VEGA62. ....... 24
26 Comparison of results of tests VEGA51, VEGA52 and VEGA62 (zoom). ....... 25
1 Introduction

This report presents the implementation in the EUROPLEXUS code of the possibility of storing a solution for later re-mapping it as initial conditions for a subsequent simulation.

EUROPLEXUS [1] (also abbreviated as EPX) is a computer code jointly developed by the French Commissariat à l’Energie Atomique (CEA DMT Saclay) and by EC-JRC. The code application domain is the numerical simulation of fast transient phenomena such as explosions, crashes and impacts in complex three-dimensional fluid-structure systems. The Cast3m [2] software from CEA is used as a pre-processor to EPX when it is necessary to generate complex meshes.

The interest for a general solution re-mapping algorithm in a code such as EPX is evident, since such an algorithm would allow to perform complex simulations that would be impossible or impractical (e.g. due to high CPU cost) to carry out as a monolithic calculation.

A fully general remapping algorithm may be extremely complex to set up, especially in a code such as EPX which contains a lot of complicated models and modelling possibilities. However, even a more modest algorithm with some limitations could be very useful in practice.

The present work was stimulated by the collaboration between JRC and NTNU about the simulation of complex shock tube experiments performed in a dedicated facility in Trondheim. A shock tube is a very long facility consisting of three main parts: i) a driver section, which is initially pressurized and contains also some firing chambers separated by membranes; ii) the shock tube proper, consisting of a rather long (~16 m) constant-section rigid-wall tube where the shock wave is formed and propagated; and iii) the test section, where the specimen to be studied under shock loading is placed.

A typical experiment may take about 30 ms to release and propagate the shock wave until it reaches the specimen, which thereafter fails in just one or two ms. Thus, in a monolithic simulation the majority of the CPU time would be spent in the first (and less interesting, albeit fundamental) phase of the calculation. Hence the idea of splitting the simulation into two parts.

During systematic test campaigns, often the first phase (wave generation and transmission) is kept constant, and only the specimen is changed. It is therefore a waste of CPU time to repeat the full simulation for each experiment. The transmission phase is independent of the particular specimen chosen and may (should in fact) be simulated only once. The mapping algorithm should then be able to store the solution at the end of the transmission phase. Then, the solution would be mapped on the same or on a different mesh as initial condition for a subsequent simulation where a specific test specimen is inserted. The mapping strategy then consists of two parts:

- A command to generate and store the map file during the first simulation.
- A command to read back the previously stored map file and to use as initial conditions for a subsequent simulation.

The two commands are described in the next Section.

It should be noted that prior to this work EPX already offered a mapping technique called “blast mapping” and represented by the MAPB command described in [3]. The two mapping strategies are
similar (they both involve two phases) but the MAPB command is dedicated to the particular case of 1D to 2D or 1D to 3D mapping under perfectly spherical blast conditions (in the first phase.) Therefore, only a 1D solution may be used in the current implementation of the MAPB command.

The present MAPP command is potentially more general since it allows mapping from a full 3D model, although with some limitations in this first implementation (see Section 2.3). Hopefully the present mapping model will become more (if not fully) general in forthcoming developments.

2 Mapping commands

We now describe the newly developed commands for the mapping.

2.1 The ECRI FICH MAPP directive

The new command to generate the map file is part of the ECRI directive:

```
ECRI ... FICH <FORM> <SPLI> MAPP <nmapp> OBJE /LECT/ /CTIM/
```

- The optional FORM keyword creates a formatted (ASCII) map file, instead of the default unformatted (binary) map file.
- The optional SPLI keyword splits the map file into several files in case the map has to be written at more than one time station. Such files are automatically named <basename>0001.map, <basename>0002.map etc., where <basename> is the base name of the EPX input file.
- The optional nmapp is the number of the logical unit of the map file or the file name in quotes. If omitted, the program chooses a file name by default. The default extension is .MAP.
- The OBJE keyword introduces the list of elements (specified by the following /LECT/) whose solution must be stored on the map file for subsequent mapping on an equivalent (or a different) mesh.
- Finally, the /CTIM/ directive allows to choose the time station(s) at which the mapping file should be produced.

2.2 The INIT MAPP directive

The new command to read back a map file and to use it for initialization of the second calculation is part of the INIT directive:

```
INIT ... MAPP <FORM> <nmapp> <MATC> OBJE /LECT/
```

- The optional FORM keyword specifies that the map file is formatted (ASCII.) By default, an unformatted (binary) map file is assumed.
- The optional nmapp is the number of the logical unit of the map file or the file name in quotes.
- The optional MATC keyword declares (under the user's responsibility) that the target object perfectly matches the source object, i.e. that the two objects are composed by the same elements and by the same nodes (albeit perhaps with different element and node indexes.) This option greatly facilitates and speeds up the solution mapping (since it requires no interpolation, but only a relatively simple search) and should be used whenever appropriate.
- The OBJE keyword introduces the list of elements (specified by the following /LECT/) of the current model onto which the solution from the map file should be mapped.
2.3 Limitations

Although it has been designed for full generality, the mapping algorithm is subjected to some limitations in its current implementation:

- The mapping can only affect the fluid sub-domain. In other words, both the source object and the target object must be composed exclusively of fluid Finite Elements and of fluid Finite Volumes of the CCFV family, i.e. cell-centred finite volumes. No node-centred Finite volumes (of the NCVF family) and no structural elements are treated at the moment. Both 2d-3D and 1D VFCCs are allowed in the same calculation.
- The nodes of the two objects must be Eulerian (i.e., fixed in space).
- The spatial dimension of the two objects must be the same.

At least some of these limitations will be progressively removed as part of forthcoming developments.

3 Implementation notes

The new generic mapping algorithm is mainly implemented in a new module M_MAPPING. Besides the usual “service” routines typical of any module (initialization and destruction of the data structure, reading of the input commands, etc.) the module features three main routines:

- WRITE_MAPPING generates and writes the mapping file during the first run. This routine is called from IMPSOR (or from IMPSOR_MPI in parallel calculations.)
- READ_MAPPING reads back the mapping file and stores it in a dedicated data structure which will then be used in the initialization phase of the calculation, after the entire input data set has been read. This routine is called from INIT, the routine which reads the INIT (initial conditions) directive.
- APPLY_MAPPING uses the previously read back mapping data to (re-)initialize the starting conditions for the second run. This routine is called by INITIA.

One of the major tasks of the mapping strategy is to search for a correspondence between the mesh used in the first run (more precisely, the part of that mesh stored in the map file), called the source mesh, and the mesh used in the second run (again, the part of that mesh that the user declares as affected by the mapping), or target mesh.

3.1 Data structure

The data structure, besides some scalars, consists mainly of a series of arrays. We will refer to these as either non-flattened or flattened arrays. A non-flattened array refers to the original numbering of elements and nodes in the corresponding mesh (of either the first or a subsequent run.) Since the data concern only a subset of the meshes, non-flattened arrays typically contain holes in their numbering. A flattened array, instead, refers only to the subset of elements and nodes used for the mapping and therefore contains no holes.

The most important non-flattened arrays are:

- MAP_FROM(1:N_MAP_FROM) contains the list of the N_MAP_FROM elements forming the source mesh (original numbering in the first simulation.)
- MAP_TO(1:N_MAP_TO) contains the list of the N_MAP_TO elements forming the target mesh (original numbering in the second simulation.)
- VFCC_MAP_FROM(1:N_VFCC_MAP_FROM) contains the list of the N_VFCC_MAP_FROM VFCCs in the source mesh (original numbering in the first simulation.)
VFCC_1D_MAP_FROM(1:N_VFCC_1D_MAP_FROM) contains the list of the N_VFCC_1D_ MAP_FROM one-dimensional VFCCs (1D-VFCCs) in the source mesh (original numbering in the first simulation.)

VFCC_MAP_TO(1:N_VFCC_MAP_TO) contains the list of the N_VFCC_MAP_TO VFCCs in the target mesh (original numbering in the second simulation.)

VFCC_1D_MAP_TO(1:N_VFCC_1D_MAP_TO) contains the list of the N_VFCC_1D_MAP_TO one-dimensional VFCCs (1D-VFCCs) forming the source mesh (original numbering in the second simulation.)

The most important flattened arrays are:

- MAP_NUMN(:) contains the connectivity (list of the nodes) of the N_MAP_FROM elements forming the source mesh.
- MAP_INDOX(1:N_MAP_FROM,1:3) contains the characteristics of the N_MAP_FROM elements forming the source mesh.
- MAP_XINIT(1:MAP_IDIM,1:N_MAP_FROM) contains the initial coordinates of the nodes forming the source mesh.
- MAP_POSECR(1:N_MAP_FROM) contains the pointer into the internal variables of the elements forming the source mesh.
- MAP_ECR(:) contains the list of the internal variables of the elements forming the source mesh.

Other important arrays are:

- IN_SOURCE_ELEM(1:N_MAP_TO) contains the correspondence between the target and the source elements. That is, IN_SOURCE_ELEM(I) is equal to IEL_LOC (ranging from 1 to N_MAP_FROM) if the I-th target element centroid lies within the IEL_LOC-th source element. Otherwise, IN_SOURCE_ELEM(I) is equal to 0 (no correspondence) and an error message is raised if the MATC keyword was specified.

- IN_SOURCE_ELNOD(1:N_NMAP_TO) contains the correspondence between the target and the source nodes. That is, IN_SOURCE_ELNOD(J) is equal to IND_LOC (ranging from 1 to N_MAP_FROM) if the J-th target node lies "exactly" (within a tiny tolerance) on the IND_LOC-th source node. It is equal to −IND_LOC (from −N_MAP_FROM to −1) if the J-th target node does not lie on any source node but it lies within the IND_LOC-th source element (an error message is raised if the MATC keyword was specified.) Otherwise, IN_SOURCE_ELNOD(J) is equal to 0 (no correspondence) and an error message is raised if the MATC keyword was specified.

### 3.2 Fast search of the mapping correspondence

The search for correspondence used to build up the data structure described in the previous paragraph is performed by a fast algorithm (bucket sorting.) The search algorithm is implemented in module M_FAST_SEARCH_GENERIC, which uses module M_MINMAXCOORD_GENERIC to build up the bounding box and the centroid of the elements involved.

The difference between M_MINMAXCOORD and M_MINMAXCOORD_GENERIC is that the former operates on all elements of the current mesh while the second operates only on the set of elements specified by the programmer when invoking the module’s services, and is therefore more flexible.

The fast search grid is built automatically and in the current implementation no parameters associated to this grid are accessible to the user via input commands.

Note that all correspondences are established based on the *initial* configuration of both the source (MAP_XINIT) and the target (XINIT) mesh. The distinction between initial and current positions is irrelevant as long as all nodes involved are Eulerian (fixed in space), as it is assumed in the current implementation. However, this might become important when this limitation is removed in a forthcoming development.
4 Numerical examples

In this Section we present some numerical examples illustrating the use of the mapping algorithm.

4.1 Simple 2D tests

We start by some simple 2D academic examples just to test the correctness of the mapping algorithm in cases with matching meshes. The calculations performed are summarized in Table 1.

<table>
<thead>
<tr>
<th>Test name</th>
<th>Comments</th>
<th>Final time [ms]</th>
<th>Steps</th>
<th>CPU [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAPP00</td>
<td>Reference solution, monolithic run</td>
<td>80.0</td>
<td>69</td>
<td>0.05</td>
</tr>
<tr>
<td>MAPP01</td>
<td>First run, write map file at 50 ms</td>
<td>80.0</td>
<td>69</td>
<td>0.06</td>
</tr>
<tr>
<td>MAPP02</td>
<td>Second run, read map file at 50 ms</td>
<td>80.0</td>
<td>27</td>
<td>0.05</td>
</tr>
<tr>
<td>MAPP03</td>
<td>First run, add leading element</td>
<td>80.0</td>
<td>69</td>
<td>0.05</td>
</tr>
<tr>
<td>MAPP04</td>
<td>Second run, add trailing element</td>
<td>80.0</td>
<td>27</td>
<td>0.05</td>
</tr>
<tr>
<td>MAPP05</td>
<td>Second run, random numbering</td>
<td>4.0</td>
<td>2</td>
<td>0.00</td>
</tr>
<tr>
<td>MAPP06</td>
<td>Second run, random numbering</td>
<td>4.0</td>
<td>1</td>
<td>0.03</td>
</tr>
</tbody>
</table>

Table 1: Simple 2D numerical simulations.

4.1.1 Case MAPP00

This test is a simple ideal shock tube, discretized by 100 VFCCs in 2D (Q4VF elements), as shown in Fig. 1. The mesh definition is intentionally embedded in the EPX file so that the numbering of elements and nodes is perfectly regular.

![Figure 1: Geometry of test case MAPP00.](image)

The results of this calculation will be used as a reference solution for the subsequent simulations using the mapping algorithm. The results in terms of fluid pressure and fluid density, represented by spatial distributions along the tube every 10 ms until 80 ms, are shown in Fig. 2.

Note that in this calculation, as well as in the following ones for this Section, the `OPTI STEP IO` command [1] is specified in the input file. This produces outputs (in particular the ALIC output file used for the post-processing) at the precisely chosen time stations (in this case exactly each 10.0 ms) rather than at approximate times. The code slightly adapts the time step in order to exactly the chosen time stations exactly. This will allow a perfectly accurate comparison of results (and the...
4.1.2 Case MAPP01

This is the first run of a simulation with mapping. The input is exactly the same as for the reference case MAPP00 but we add the request for the creation of a map file at 50.0 ms:

```
ECRI VFCC TFRE 10.E-3
FICH ALIC TFRE 10.E-3
FICH FORM MAPP OBJE LECT tous TERM TIME PROG 50.E-3 TERM
```

The map file will be formatted (FORM) and will use the default name (here mapp01.map). All the elements in this mesh, in this case all 100 VFCCs, will be stored in the map file.

Results of this calculation (up to 80 ms) are identical to those of the reference case MAPP00, of course, and are not shown for brevity. The only difference with respect to MAPP00 is that a map file mapp01.map is written on the current directory. The contents of the map file (a pure text file in this case) is similar to the following snippet:

```
EUROPLEXUS GENERIC MAP FILE GENERATED ON 15/03/2018
STEP 42 TIME 5.000000000000E-02
IDIM 2
N_MAP_FROM : 100 LEN_NUMN : 400
ELEMENT TYPE MTYP / NODES
1 132 9
1 2 103 102
2 132 9
2 3 104 103
3 132 9
3 4 105 104
. . .
IEL: 100 IVFCC 100
1.000000000000E+00 1.000000000000E+00 1.000000000000E+00 0.000000000000E+00
1.797693134862E+302 1.000000000000E+00
4 SOL_UCONS_VFCC
1.300000000000E+00 0.000000000000E+00 0.000000000000E+00 2.487562189055E+05
0 SOL_UCONS_OLD_VFCC
4 SOL_UCONS_INT_VFCC
0.000000000000E+00 0.000000000000E+00 0.000000000000E+00 0.000000000000E+00
0 UTRANS_VFCC
0 UFLUID_VFCC
1 UDEP_VFCC
1.000000000000E+05
0 UPRIM_VFCC
0 UPRIM_VFCC
0 SOURCE_VFCC
0 T_UCLS
0 REST_VFCC
N_VFCC_1D_MAP_FROM : 0
```

4.1.3 Case MAPP02

This is the second run of a simulation with mapping. We use exactly the same mesh (same elements and nodes, same numbering) as in case MAPP01 for simplicity in this first example. The INIT MAPP
command is used to read back the map file:

```
INIT MAPP FORM 'mapp01.map' MATC OBJE LECT tous TERM
```

We must declare the map file as formatted here (FORM) and specify its name (‘mapp01.map’), otherwise the name by default would be used (‘mapp02.map’), which does not exist. As target mesh we declare all elements, for simplicity in this first example.

Note that we declare that the target mesh matches the source mesh (MATC keyword.) The code then assumes that to each target element there must correspond one and only one source element (the comparison is based upon the positions of the element centroids), and that to each target node there must correspond one and only one source node (the comparison is based upon the positions of the nodes.) The node and element connectivity, i.e. the actual numbering of nodes and elements, can be different in the two meshes. A small tolerance is used in comparing centroid and node positions because, when using a formatted map file, the source coordinates cannot be transferred with full (double) precision via the map file. The tolerance used is 10 times the smallest real (Fortran single precision) number that can be represented by the machine, which should roughly corresponds to 6 significant digits on most computer architectures.

The code reads back the map file, sets the initial time of the calculation to the value (50.0 ms) that has been found on this file (thus overriding the TINI value contained in the EPX input file), maps the source solution from the map file to the target as “initial” conditions and then performs the simulation until the chosen final time of 80.0 ms.

The results of this calculation (black curves) are compared in Fig. 3 against the results of case MAPP01 (red curves), for the time instants between 50.0 and 80.0 ms (since, obviously, only these instants are available in the results file of the second run), showing excellent agreement. Each couple of curves is perfectly superposed so that only the red curves are visible.

![Figure 3: Comparison of results of tests MAPP01 and MAPP02 for $t \geq 50$ ms.](image)

**4.1.4 Cases MAPP03 and MAPP04**

These calculations are a repetition of cases MAPP01 and MAPP02, respectively, where we use a slightly different mesh. In case MAPP03 we add a leading node and a leading element (a fake PMAT) while in case MAPP04 we add a trailing node and a trailing element (also a fake PMAT.) In this way, the numbering of both elements and nodes is altered with respect to the previous cases, while still conserving perfect geometrical matching of the source and target meshes in terms of coordinates.

In the first run we must identify the fluid zone, that we want to use for the mapping (we can no longer use tous to choose all elements):

```
COMP GROU 4 'pm' LECT 1 TERM
'hp' LECT 2 PAS 1 51 TERM
```
And similarly in the second run we have:

```
COMP GROU 4 'hp' LECT 1 PAS 1 50 TERM
'lp' LECT 51 PAS 1 100 TERM
'fl' LECT 1 PAS 1 100 TERM
'pm' LECT 101 TERM
```

The two solutions are compared in Fig. 4 and are in excellent agreement.

Figure 4: Comparison of results of tests MAPP03 and MAPP04 for \( t \geq 50 \text{ ms} \).

### 4.1.5 Cases MAPP05 and MAPP06

These tests use an extremely simple mesh with only 5 Q4VF VFCCs. However, the two meshes are spatially “offset” with respect to each other by one element, so that only four of the elements in each mesh do have a matching element in the other mesh. In addition, the connectivity (i.e. the numbering of nodes and elements) is completely random and different in the two cases. The models are shown in Fig. 5, including also the reference frame in order to highlight the offset between the two meshes.

The complete input files are compared side by side below:
Note that the two objects (source and target) used for the mapping involve only three of the four “common” elements to the two meshes. These are the elements highlighted in yellow in Fig. 5. Since these two mesh portions are fully matching as far as concerns the element centroids and the nodal coordinates (albeit with completely different numberings) we can still declare the target mesh as matching (MATC keyword.)

After running the two tests, it is verified that the source solution (which had been stored at step 1) is correctly mapped onto the target mesh.

4.2 Simple 3D tests

We now consider some simple 3D academic examples. The problem studied is the ideal shock tube already considered in Section 4.1. The calculations performed are summarized in Table 2.

<table>
<thead>
<tr>
<th>Test name</th>
<th>Comments</th>
<th>Final time [ms]</th>
<th>Steps</th>
<th>CPU [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAPP09</td>
<td>Reference solution, monolithic run</td>
<td>80.0</td>
<td>70</td>
<td>0.08</td>
</tr>
<tr>
<td>MAPP07</td>
<td>First run, write map file at 50 ms</td>
<td>80.0</td>
<td>70</td>
<td>0.09</td>
</tr>
<tr>
<td>MAPP08</td>
<td>Second run, read map file at 50 ms</td>
<td>80.0</td>
<td>27</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Table 2: Simple 3D numerical simulations.

4.2.1 Case MAPP09

This is the monolithic 3D solution of the shock tube problem, using 100 8-node CUVF finite volumes. In addition to the change in spatial dimension (from 2 to 3), another difference of this test with respect to case MAPP00 is that here we use a full second-order in space and time solution for the VFCCs, while in case MAPP00 it was a first-order in space and time solution:
The results in terms of fluid pressure and fluid density, represented by spatial distributions along the tube every 10 ms until 80 ms, are shown in Fig. 6.

Fig. 7 compares the solutions MAPP00 and MAPP09. The change in spatial dimension has no effect (as verified on other occasions) so the differences are due to the second order scheme used in the latter solution.

4.2.2 Cases MAPP07 and MAPP08

These tests show the split solution of the 3D shock tube problem (first and second run, respectively.) The results of test MAPP07 are of course identical to those of test MAPP09 (since we have only added the creation of the map file at 50 ms) and are not shown for brevity.

The results of case MAPP08 (black curves) are compared in Fig. 8 against the results of case MAPP07 (red curves), for the time instants between 50.0 and 80.0 ms, showing excellent agreement: the curves are superposed and only the red ones are visible.
4.3 Simple 1D tests

We now consider some simple 1D academic examples equivalent to the 2D examples presented in Section 4.1. The calculations performed are summarized in Table 3.

<table>
<thead>
<tr>
<th>Test name</th>
<th>Comments</th>
<th>Final time [ms]</th>
<th>Steps [s]</th>
<th>CPU [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAPP10</td>
<td>Reference solution, monolithic run</td>
<td>80.0</td>
<td>69</td>
<td>0.05</td>
</tr>
<tr>
<td>MAPP11</td>
<td>First run, write map file at 50 ms</td>
<td>80.0</td>
<td>69</td>
<td>0.06</td>
</tr>
<tr>
<td>MAPP12</td>
<td>Second run, read map file at 50 ms</td>
<td>80.0</td>
<td>27</td>
<td>0.05</td>
</tr>
<tr>
<td>MAPP17</td>
<td>First run, second order</td>
<td>80.0</td>
<td>69</td>
<td>0.03</td>
</tr>
<tr>
<td>MAPP18</td>
<td>Second run, second order</td>
<td>80.0</td>
<td>27</td>
<td>0.03</td>
</tr>
</tbody>
</table>

Table 3: Simple 1D numerical simulations.

4.3.1 Case MAPP10

This is a monolithic run, equivalent to case MAPP00 presented previously but using the 1D mesh (100 TUVF volumes) shown in Fig. 9.

The results obtained are shown in Fig. 10 and are practically identical to those obtained in 2D, see Fig. 2.

4.3.2 Cases MAPP11 and MAPP12

These tests are similar to cases MAPP01 (first run) and MAPP02 (second run), respectively, but use the same 1D mesh as case MAPP10. Inputs (apart from the mesh definition) are practically identical to those of tests MAPP01 and MAPP02 and are not commented for brevity.

The results of tests MAPP11 (in red) and MAPP12 (in black) for \( t \geq 50 \) ms are compared in Fig. 11, showing excellent agreement (red curves superposed to the black curves).

4.3.3 Cases MAPP17 and MAPP18

These tests are similar to cases MAPP11 (first run) and MAPP12 (second run), respectively, but use second order in space and time finite volumes, like in cases MAPP07 and MAPP08.

No separate monolithic solution is previously obtained, unlike in previous examples. In fact, the solution MAPP17 is pursued until the final time of 80.0 ms and can be considered as a monolithic solution (reference), with the only addition of the creation of a map file which, as verified in the previous examples, has no influence on the solution itself. The results for \( t \geq 50 \) ms are compared in Fig. 12, showing excellent agreement (superposed red and black curves).
By comparing the results of case MAPP17 (second order) against those of case MAPP11 (first order), one can see no differences. In fact, it turns out [1] that, at the moment of this writing, second order in space and in time is not currently implemented in 1D VFCCs, so the ORDR and OTPS optional keywords are simply ignored by any 1D-VFCCs present in a calculation (however, these options do have an effect on 2D-3D VFCCs).

4.4 Simple combined 1D/3D tests

We now want to perform some simple tests containing a mixture of 3D and 1D finite volumes, which is more general and complex than having only either of the two types of volumes in a calculation. This will be also the case in the target NTNU shock tube applications.

The calculations performed are summarized in Table 4.

4.4.1 Case MAPP20

This test is inspired by case 1D3D14 of reference [4]. A shock tube is modelled, from left to right, by a 3D part (1200 CUVF), followed by a 1D part (1000 TUVF), followed again by a 3D part (1000...
CUVF), as shown (not in scale) in the scheme of Fig. 13. All elements have a uniform size of 1 cm.

In order to connect the various parts of the model, two TUBM elements are used at the junctions between zones of different spatial dimension, see [4] for details. Unlike all previous test cases, the mesh is generated by Cast3m. This can at times lead to a pseudo-random numbering of elements and (in particular) of nodes. It is important to check that the mapping algorithm works well under such conditions.
In this test we obtain a monolithic solution, identical to that of [4], to serve as a reference. The solution in terms of pressure and density distributions along the tube at the final chosen time of 20.0 ms is presented in Fig. 14. The black curves are the numerical results and the red curves are the analytical solutions, which are also shown for comparison.

Figure 14: Results of test case MAPP20.

4.4.2 Case MAPP21

This test is the first run of a split calculation. The input is identical to that of case MAPP20 but we add the request to generate a map file at 10 ms:

```plaintext
ECRI ECRO VFCC TFRE 1.E-3
! NOPO NOEL
FICH ALIC TFRE 1.E-3
FICH FORM MAPP OBJE LECT hp3d lp3d hp1d lp1d TERM
TIME PROG 10.E-3 TERM
```

Note that we specify both the 3D VFCCs (hp3d, lp3d) and the 1D VFCCs (hp1d, lp1d) in the OBJE definition, but we tentatively do not include the junction (raccord) elements TUBM. It will have to be verified a posteriori that this has no effect on the (second run) solution and is therefore actually unnecessary.

The results of this calculation are (obviously) identical to those of case MAPP20 and are not shown for brevity.

4.4.3 Case MAPP22

This is the second run. Again, the input is identical to that of case MAPP20 except for the command which triggers re-initialization from the map file:

```plaintext
INIT MAPP FORM 'mapp21.map' MATC OBJE LECT hp3d lp3d hp1d lp1d TERM
```

The results of this test (black curves) at the final time \( t = 20.0 \) ms are compared in Fig. 15 against those obtained in test MAPP21 (green curves), showing excellent agreement. This confirms that, indeed, it is not necessary to include the junction elements TUBM in the mapping file.

4.5 3D tests

We now consider tests involving a slightly more complex geometry than a simple shock tube. The calculations performed are summarized in Table 5.
Figure 15: Comparison of results of tests MAPP21 and MAPP22 at $t = 20$ ms.

<table>
<thead>
<tr>
<th>Test name</th>
<th>Comments</th>
<th>Final time [ms]</th>
<th>Steps</th>
<th>CPU [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAPP30</td>
<td>Reference solution, monolithic run</td>
<td>30.00</td>
<td>49</td>
<td>0.08</td>
</tr>
<tr>
<td>MAPP31</td>
<td>First run, write map file at 3 ms</td>
<td>30.00</td>
<td>4</td>
<td>0.02</td>
</tr>
<tr>
<td>MAPP32</td>
<td>Second run, read map file at 3 ms</td>
<td>30.00</td>
<td>45</td>
<td>0.09</td>
</tr>
</tbody>
</table>

Table 5: More complex 3D numerical simulations.

4.5.1 Case MAPP30

This test studies an explosion in a square parallelepiped of fluid measuring $10 \times 10 \times 1$ units. This first run is used to obtain the monolithic reference solution and uses the complete model, shown in Fig. 16. The explosive (in red) is located at one corner of the fluid domain, whose walls are considered as rigid.

The fluid pressure is recorded at the four corners of the fluid domain. They are shown in Fig. 17. The left part of the Figure shows all four pressure records. In the right part of the Figure, the bomb pressure record is removed so that the scale is more adequate for the other records.

From this solution, and also by visualizing pressure distributions at the various time steps (not shown for brevity), we find out that until 3.0 ms (step number 4 in this simulation) the perturbation caused by the bomb explosion remains confined within the “inner” $5 \times 5 \times 1$ sub-domain of the parallelepiped, that is the left-bottom quadrant of the parallelepiped, containing the bomb. The fluid conditions in the rest of the fluid domain are still completely unperturbed at this time.
4.5.2 Case MAPP31

This is the first run of a split simulation. The mesh, shown in Fig. 18, includes only the left-bottom quadrant of the parallelepiped, containing the bomb. A map file is written at 3.0 ms and then the simulation is stopped. The entire fluid domain (i.e. the entire $5 \times 5 \times 1$ units fluid block) is designated as the source object for the mapping.

The mesh is generated by Cast3m and therefore the numbering of elements and nodes is substantially random, and radically different from the one (also made by Cast3m) of test MAPP30.

4.5.3 Case MAPP32

This is the second part of the split run. The mesh includes the entire $10 \times 10 \times 1$ fluid domain, like in the monolithic solution MAPP30. The map file from the previous simulation is read back and used to re-initialize the solution (at $t = 3.0$ ms) in the $5 \times 5 \times 1$ fluid sub-domain. The calculation is then continued until 30 ms.

The results of this simulation (solid curves) are compared to those of the monolithic solution MAPP30 (dashed curves) in Fig. 19. They are virtually superposed, thus confirming the correct performance of the split solution strategy in this example.
### 4.6 NTNU shock tube tests

We will now apply the mapping model to some more realistic tests simulating the actual NTNU shock tube and described in reference [5]. In fact, the main goal of the present work was to provide a way to speed up and rationalize sets of parametric calculations like the ones presented in [5], by means of a calculation splitting technique. The calculations performed are summarized in Table 6.

The first two cases (VEGA51 and VEGA52) follow the same procedure used in the previous academic tests: a first complete monolithic run to obtain a complete reference solution and to produce a map file at an intermediate time, followed by a second run starting from the map file and computing the solution until the final time. This procedure allows thorough comparison of results and validation of the mapping technique, but is penalizing in terms of CPU time.

The second set of simulations (VEGA61 and VEGA62) illustrates a more typical and more CPU-efficient way of using the mapping technique in real practice. The first run uses only (the strictly necessary) part of the complete model, produces a map file at a certain intermediate time and then stops immediately. The second run uses the part of the complete model necessary for the second phase (which is different from the one used in the first phase), reads back the map file and runs the simulation until the final time. So, neither calculation ever computes the complete model that would be used in a monolithic simulation (such as VEGA51). Also, the time intervals of the two parts of the simulation are completely disjoint.

<table>
<thead>
<tr>
<th>Test name</th>
<th>Comments</th>
<th>Initial time [ms]</th>
<th>Final time [ms]</th>
<th>Steps</th>
<th>CPU [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>VEGA51</td>
<td>Reference solution, monolithic run</td>
<td>0.0</td>
<td>40.0</td>
<td>110 451</td>
<td>155 354</td>
</tr>
<tr>
<td>VEGA52</td>
<td>Second run, from map file</td>
<td>28.0</td>
<td>40.0</td>
<td>85 196</td>
<td>107 151</td>
</tr>
<tr>
<td>VEGA61</td>
<td>First run, write map file at 28 ms</td>
<td>0.0</td>
<td>28.0</td>
<td>23 627</td>
<td>20 389</td>
</tr>
<tr>
<td>VEGA62</td>
<td>Second run, read map file at 28 ms</td>
<td>28.0</td>
<td>40.0</td>
<td>85 196</td>
<td>107 365</td>
</tr>
<tr>
<td>VEGA53</td>
<td>Same as 51 but no STEP 10</td>
<td>0.0</td>
<td>10.0</td>
<td>22 271</td>
<td>41 325</td>
</tr>
<tr>
<td>VEGA63</td>
<td>Same as 61 but no STEP 10</td>
<td>0.0</td>
<td>10.0</td>
<td>20 322</td>
<td>20 023</td>
</tr>
</tbody>
</table>

Table 6: NTNU shock tube simulations.

#### 4.6.1 Case VEGA51

This is a monolithic simulation in order to obtain a complete reference solution. The test is inspired by case VEGA45 of reference [5]. The mesh is identical to that of case VEGA45 and, as concerns the EPX input file, the following minor modifications are applied:

- The final time of the simulation is reduced from 80 ms to 40 ms.
- Fewer printouts and ALIC storages are requested.
- The **STEP IO** option is activated in order to achieve exact storage times (for an exact verification of the mapping).
- The production of a map file is requested at \( t = 28 \) ms:

```
ECRI . . .
FICH FORM MAPP OBJS LECT flui3d tubelp1 TERM
TIME PROG 28.0E-3 TERM
```

Based on results of the previous calculation VEGA45 of reference [5], we verified that at the chosen time for the map file creation \( (t = 28 \) ms) the shock wave has not yet reached the end of the 1D part of the shock tube model. The fluid in the 3D test region of the device and the specimen are still at complete rest at this time. So, as a general rule, the model part to be chosen for the mapping (flui3d and tubelp1) should contains all the parts of the model (more precisely, of the fluid model) that have been perturbed at the chosen mapping time.

A very important condition to be verified is that, at the moment of writing the mapping, the concerned parts of the model (here flui3d and tubelp1) are not currently in a refined adaptive state. In fact this first implementation of the mapping model cannot deal with adapted meshes.

More precisely, the mapped model parts may have been subjected to adaptivity at earlier (or later) times (and this is indeed the case for the fluid near the membranes in the present example), but the mesh must be in a completely un-refined state when the map file is written. This condition is satisfied in the present example because we completely erode the membranes (more precisely, the still surviving membrane fragments) at \( t = 10 \) ms by the **OPTI FANT** directive and, as a consequence, the fluid in the mapped part of the device is guaranteed to be completely un-refined at \( t = 28 \) ms when writing the map file.

Some results of this calculation are presented next. Fig. 20(a) shows the fluid pressures at sensors S1 and S2, while Fig. 20(b) shows the central displacement of the plate.

![Figure 20: Results of test VEGA51.](image)

4.6.2 Case VEGA52

This test is a second calculation starting from the map file produced by test VEGA51 at \( t = 28.0 \) ms. The mesh file is the same as in case VEGA51 (complete model) but we remove (Gibiane *oub1* command) the membranes (*mems*), the associated pressure elements (*pre*) and any related models both from the Cast3m input file:
and from the EPX file. Note also how the mesh object is reconstructed from the remaining parts.

Some results of this calculation are presented next. Fig. 21(a) shows the fluid pressures at sensors S1 and S2, while Fig. 21(b) shows the central displacement of the plate.

Fig. 22 compares the results of cases VEGA51 (dashed lines) and VEGA52 (solid lines). The curves are superposed so we may conclude that the mapping algorithm works well in this case. Fig. 23 presents the same comparison, but over a much shorter time window (between 28 and 33 ms), so as to better appreciate the correctness of the results.

### 4.6.3 Case VEGA61

This is a first run of the shock tube problem where, for efficiency, we include only the driver, the firing sections with the membranes, and the entire 1D part of the tube. The test section is not present. We
Figure 23: Comparison of results of tests VEGA51 and VEGA52 (zoom).

write a map file at $t = 28$ ms and then we immediately stop the calculation.

The mesh is similar to that of test case VEGA51 but we remove (OUBL Gibiane command) all the unnecessary geometrical objects definitions:

```
opti rest form 'vega33.msh';
rest form;
list;
oubl mesh;
oubl spec;
oubl stril;
oubl equa;
oubl eprid;
oubl ecubd;
oubl mplate;
oubl presur;
oubl iframe;
oubl preplat;
oubl urame;
oubl devi;
oubl flai;
oubl tube;
oubl raclep;
oubl tubelp;
oubl tubelp3;
oubl abso;
utb tank;
utb stub3d;
utb mesh1;
utb iface3d;
utb hout;
utb fonde;
utb p3hd;
utb gyy;
utb pgg;

*;
list;
mesh = flui3d et tubelp1 et rac3d1d et face3d et mems et pre;
tass mesh noop;
opti sauv form 'vega61.msh';
sauv form mesh;
*
opti trac psc ftrc 'vega61_mesh.ps';
trac cach mesh;
*
fin;
```

The EPX input file is also simpler than that of case VEGA51. Note the command to produce the map file:

```
VEGA61
ECHO
!CONV win
CAST mesh
THD ALE
ENED 1.0
DIME ADAP RPO1 100000
Q4GS 20000
CSVF 150000
NWF1 200000
CLNS 20000
NPSI 20000
ENDA
JOSA 475 ! Total n. of nodes in a TUBM junction
NALE 1 NBLE 1
TERM
GEOM Q4GS mems
CSVF flu3d
TVPF tub3d1
CLNS pre face3d
TERM rac3d1d
TERM
COMP EPAI 1.50E-3 LECT mems TERM
DIAM 0.1692568 LECT tubelp1 TERM
FACE LECT face3d TERM CHEF 1.0
GROUP 4 'face1' LECT flu3d TERM
COND XE QT -16.405
COND XB LT -16.005
!'S1' LECT tube TERM COND NEAR POIN -0.246 0.15 0.15
!'S2' LECT tube TERM COND NEAR NODE LECT pla TERM
!'spar2' LECT tubelp1 TERM COND NEAR NODE LECT pla TERM
!'tubelp' LECT tubelp1 DIFF spar2 TERM
TERM ? 'memm' LECT mems TERM
!
COUL THQ LECT tubelp1 tra TERM
VERE LECT fir2 TERM
RESQ LECT driver TERM
NVSQ LECT driver TERM
GRIL LECT mems TERM
JAUN LECT pre TERM
ADAP THQ ECO3 3 TMIR 0.01 MAX 0.4 MAX 3
LECT mems TERM
GRIL LECT mems TERM
FONC 1 TABL 5 0.0 0.0
1.0E-3 1.0
2.999E-3 1.0
3.0E-3 0.0
100.0E-3 0.0
MATE
!LOI 1
VEG3 RD 1380 YOUN 2757.985 NU 0.495 ELAS 120.66 ! "Melinex/Mylar/PET"
FAIL PEPT LMQ 1.0
TRAC 3 120.66 0.04351
22
No results are presented for this test because the plate was not present and the S1 and S2 pressure sensors are not yet hit by the pressure wave at the chosen mapping (final, in this case) time of 28 ms.

4.6.4 Case VEGA62

This test is a second calculation starting from the map file produced by test VEGA61 at $t = 28.0$ ms. The mesh is the same as in case VEGA51 but without the membranes and the associated pressure elements. Therefore, we may simply re-use the Cast3m and the EPX files of test VEGA52 by just changing the calculation name and the input map file, which should be vega61.map instead of vega51.map.

Some results of this calculation are presented next. Fig. 24(a) shows the fluid pressures at sensors S1 and S2, while Fig. 24(b) shows the central displacement of the plate.
4.6.5 Comparison of results

In this Section we perform a comparison of the results of all previous calculations (whenever this is meaningful).

Fig. 25(a) compares the fluid pressures at sensor S1 while Fig. 25(b) compares the pressure at sensor S2. Finally, Fig. 25(c) compares the central displacement of the plate. The red curves correspond to test VEGA51 (i.e., to the monolithic reference solution), the green curves to case VEGA52 and the blue curves to case VEGA62.

Figure 25: Comparison of results of tests VEGA51, VEGA52 and VEGA62.

Fig. 26 presents the same results in detail over the time interval between 28 and 33 ms, which allows a better view of the (small) differences between the solutions. The solutions of cases VEGA51 and VEGA52 are in perfect agreement, while the solution of case VEGA62 appears to be shifted of about 0.05 ms, and presents also very slight differences in the peaks and oscillations. The reason for such discrepancies is under investigation.

One of the causes of the observed discrepancies could be a difference in the failure mechanism of the membranes. The membranes failure sequence in tests VEGA51 and VEGA61 is presented in Figs 27 and 28, respectively. There are some minor differences in the fragmentation and in the formation of debris.

In order to better appreciate such differences, Figs. 29 and 30 show the failure sequence of the three membranes in tests VEGA51 and VEGA61, respectively, visualized in the initial (undeformed) configuration of each membrane. The three membranes have identical dimensions but they appear to be of slightly different size in these Figures since the point of view is the same in all pictures so that the distance from the eye to the membrane increases passing from the first, to the second and to the
third membrane.

At 5 ms, only the first membrane has started to fail, while at 6 ms all three membranes have undergone some failure. Four petals are formed in the second and third membrane. In the first membrane, four main petals are formed, but an incipient failure is also observed that would lead to 8 petals if completed. As a matter of fact, in similar simulations reported in reference [5], e.g. in test case VEGA45 from which the present case VEGA51 was derived (with apparently only minor modifications in the input), eight completely formed petals are often produced in the first membrane (and only four in the other two). The reasons for this discrepancy are tentatively investigated in the next test cases.

4.6.6 Case VEGA53

By comparing the inputs of tests VEGA45 of reference [5] (8 petals in the first membrane) and the present case VEGA51 (only 4 fully formed petals in the first membrane), the only modification susceptible of modifying (though very slightly) the results is the use of the STEP IO option in the latter calculation. This option was used only to achieve a very precise comparison between solutions during the present validation process and would not be used in real applications. However, it is worthwhile to investigate its influence on the solution.

Fig. 31 shows the obtained membranes failure sequence and Fig. 32 shows the same sequence in the undeformed configuration. The result at 10 ms is missing (and therefore we plot that at 9.5 ms instead in Fig.32) simply because without the STEP IO option the code stores the results at a time slightly larger than the requested one (10.0002 ms in this case, i.e. at the first time value which is
equal to or larger than the specified one), while the membranes are eroded at (exactly) 10 ms, so at the last storage time there are no membranes any more to draw.

We can see that in this solution eight full petals are indeed formed in the first membrane, starting from 6 ms. The reason for the observed discrepancy may only reside in the time history of the simulations. Fig. 33 compares the CPU times and the time increments (stability steps) of cases VEGA53 (in black) and VEGA51 (in red). Especially the second diagram, Fig. 33(b), is interesting in the present context. We see that the history of time increments is indeed quite different in the two solutions, in the period between 4.5 and 9.5 ms.

The sharp, and substantially random-like, reductions of stability step are due to some membrane elements near the advancing crack assuming weird shapes without reaching the failure and erosion limit (which in these simulations is set to ERoS 1.0). This is unnecessarily penalizing in terms of CPU and it also has effects on the solution, as we are showing here, due to the high sensitivity of the failure and crack propagation process on the time increment used. Such elements should be eroded when they reach an unacceptable shape (e.g. based on a warping factor or aspect ratio) but this remains to be done.

4.6.7 Case VEGA63

This is a repetition of test VEGA61 but without the STEP IO option. The membranes deformations are presented in Figs. 34 and 35 and indeed, eight fully formed petals are produced in the first membrane starting at 6 ms, like in case VEGA53. Finally, Fig. 36 compares the time histories of the two simulations. The observations made in Section 4.6.6 on the influence of the time increment history are confirmed.
4.6.8 Comparison of first membrane failure patterns

The obtained failure patterns for the first membrane (in the un-deformed configuration) at 6.0 ms are summarized in Fig. 37 for direct comparison.

Figure 28: Membranes failure sequence in tests VEGA61.
Figure 29: Undeformed membranes failure sequence in test VEGA51.
Figure 30: Undeformed membranes failure sequence in test VEGA61.
Figure 31: Membranes failure sequence in tests VEGA53.
Figure 32: Undeformed membranes failure sequence in test VEGA53.
Table 3: Comparison of calculation histories of tests VEGA51 and VEGA53.

(a) Cpu time
(b) Time increment

Figure 33: Membranes failure sequence in tests VEGA63.

(a) $t = 5$ ms  (b) $t = 6$ ms  (c) $t = 7$ ms  
(d) $t = 8$ ms  (e) $t = 9$ ms
Figure 35: Undeformed membranes failure sequence in test VEGA63.
Figure 36: Comparison of calculation histories of tests VEGA61 and VEGA63.

(a) CPU time
(b) Time increment

Figure 37: Undeformed membranes failure patterns at 6 ms in tests VEGA51 through VEGA63.

(a) VEGA51, $t = 6$ ms
(b) VEGA61, $t = 6$ ms
(c) VEGA53, $t = 6$ ms
(d) VEGA63, $t = 6$ ms
5 Practical notes

The notion of \textit{time} and of \textit{time step} in calculations using the mapping algorithm deserves some explanation.

In the current implementation of the model, during a second run the code reads back the time value $t_{\text{map}}$ from the map file and uses it to reset the initial time $t_{\text{ini}}$ specified by the user by the \texttt{TINI} keyword in the \texttt{CALC} directive, i.e. it poses:

$$t_{\text{ini}} = t_{\text{map}}$$  \hfill (1)

As a consequence, the value $t_{\text{ini}}$ specified by the user is irrelevant although, for aesthetic (human input readability) reasons, it is suggested to specify a value of \texttt{TINI} (which is mandatory in any EPX input file) equal to that of the corresponding monolithic simulation (typically, but not always, equal to 0).

In this way, the times of both the first and a subsequent simulation will always coincide with the “real” (physical) times, as concerns the outputs and storages, any used time functions etc.

For the time step number, however, the convention is different. The step number is stored in the map file but it is \textit{not} reused in a second calculation. A second calculation is always set to start from step 0, like any first calculation (irrespective of the value of \texttt{TINI} chosen, either 0 or not).

This is due to the fact that the “first” (or rather the 0-th) step is special in an explicit code such as EPX. During this step, some special operations have to be performed, which are not done during the successive steps. Therefore, the code is full of tests which perform (or skip) some operations in the particular case that the step counter ($\texttt{NPAS}$) has the value 0.

Since a “second” run, in the mapping terminology, is seen as just a “normal” run by EPX, only containing some special re-initialization of physical conditions, it is essential the the step number be 0 at the beginning of such a run, otherwise one would have to change a lot of tests in the code.

For this reason, the step number restarts from 0 during a second run with the mapping model and therefore time step numbers (at generated printouts, results storages etc.) will \textit{not} correspond to those of a monolithic run (although \textit{time} values will.)

References


All the input files used in the previous Sections are listed below.

<table>
<thead>
<tr>
<th>Appendix I — Input files</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAPPO0.epx</td>
</tr>
<tr>
<td>ECHO</td>
</tr>
<tr>
<td>DCOU VDK</td>
</tr>
<tr>
<td>OPLA EULE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GEM Lib</th>
<th>P019</th>
<th>202</th>
<th>QVVF 100 TERM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
</tr>
<tr>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
</tr>
<tr>
<td>61</td>
<td>62</td>
<td>63</td>
<td>64</td>
</tr>
<tr>
<td>71</td>
<td>72</td>
<td>73</td>
<td>74</td>
</tr>
<tr>
<td>81</td>
<td>82</td>
<td>83</td>
<td>84</td>
</tr>
<tr>
<td>91</td>
<td>92</td>
<td>93</td>
<td>94</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
</tr>
<tr>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
</tr>
<tr>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
</tr>
<tr>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
</tr>
<tr>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
</tr>
<tr>
<td>70</td>
<td>71</td>
<td>72</td>
<td>73</td>
</tr>
<tr>
<td>80</td>
<td>81</td>
<td>82</td>
<td>83</td>
</tr>
<tr>
<td>90</td>
<td>91</td>
<td>92</td>
<td>93</td>
</tr>
</tbody>
</table>

**Input files**

- **MAPPO0.epx**
- **DCOU VDK**
- **OPLA EULE**
- **GEM Lib**
- **P019**
- **202**
- **QVVF**
- **100**
- **TERM**

The input files are listed in the following format:

- **MAPPO0.epx**
- **ECHO**
- **DCOU VDK**
- **OPLA EULE**
- **GEM Lib**
- **P019**
- **202**
- **QVVF**
- **100**
- **TERM**

These files contain the necessary input data for the simulations and calculations described in the document.
GEOM LIBR POIN 202 Q4VF 100 TERM
ECHO FIN
TRAC 11 12 13 14 15 16 17 18 19 AXES 1.0 'DENS [KG/M3]' SCOU 14 'r30' ECRO COMP 2 T 30.E-3 SAXE 1.0 'ABSC' LECT 1 PAS 1 101 TERM SCOU 12 'r10' ECRO COMP 2 T 10.E-3 SAXE 1.0 'ABSC' LECT 1 PAS 1 101 TERM SCOU 11 'r00' ECRO COMP 2 T 0.E-3 SAXE 1.0 'ABSC' LECT 1 PAS 1 101 TERM SCOU 7 'p60' ECRO COMP 1 T 60.E-3 SAXE 1.0 'ABSC' LECT 1 PAS 1 101 TERM SCOU 5 'p40' ECRO COMP 1 T 40.E-3 SAXE 1.0 'ABSC' LECT 1 PAS 1 101 TERM SCOU 1 'p00' ECRO COMP 1 T 0.E-3 SAXE 1.0 'ABSC' LECT 1 PAS 1 101 TERM SORT GRAP AXTE 1.0 'T [s]' RESU ALIC GARD PSCR CALC TINI 0. TFIN 80.E-3 TERM COMP GROU 2 'hp' LECT 1 PAS 1 100 TERM COUL NENG LECT hp TERM TURQ LECT hp TERM MATE GAZP RD 13. FINI 1. ES GAMI 1.402 PRED 1. ES CV 713.3 LECT hp TERM GAZP RD 1.3 PINI 1. ES GAMI 1.402 PRED 1. ES CV 713.3 LECT hp TERM ECRI FVPC TFRE 10. E-3 FICH ALIC TFRE 10. E-3 FICH FORM OBJE LECT tous TERM TIME PREG 50. E-3 TERM OPTI NOTE STEP 10 LOG 1 VFCF FODM 6 CALC TINI 0. TFIN 80. E-3

map02.epx
**Post treatment**

CALC TINI 0. TFIN 80.E-3

ECRI VFCC TFRE 10.E-3


MAPP03.exp
101 102 203 202
COMP GROUP 4: pm LECT 1 TERM
  hp1 LECT 2 PAS 1 101 TERM
  'hp' LECT 53 PAS 1 101 TERM
  f1' LECT 2 PAS 1 101 TERM
EPSI 0.01 LECT pm TERM
COOL NOIR LECT pm TERM
RUG LECT hp TERM
TURQ LECT 'hp' TERM
MATE GASP R0 13. PIRI 1.6 GAM 1.402 PRE 1.ES
CV 713.3
LECT hp TERM
GASP R0 13. PIRI 1.6 GAM 1.402 PRE 1.ES
CV 713.3
LECT hp TERM
MASS 0.0 LECT pm TERM
ECNIL VFC GASP 10. E-3
FICH ALIC GASP 10. E-3
FICH FROM GASP OBJE LECT 11 TERM TIME 50. E-3 TERM
OPTI NOTE STEP 10 LOG 1
VFC 6
mappl1.epx
mapp17.epx

MAPP17
SINO
'OCSV WIN
TRID EULE
GEOM LIBR POIN 101 TUVF 100 TERM
0 0 0 1 0 0 2 0 0 3 0 0 4 0 0 5 0 0
11 0 0 12 0 0 13 0 0 14 0 0 15 0 0
16 0 0 17 0 0 18 0 0 19 0 0 20 0 0
21 0 0 22 0 0 23 0 0 24 0 0 25 0 0
26 0 0 27 0 0 28 0 0 29 0 0 30 0 0
31 0 0 32 0 0 33 0 0 34 0 0 35 0 0
36 0 0 37 0 0 38 0 0 39 0 0 40 0 0
41 0 0 42 0 0 43 0 0 44 0 0 45 0 0
46 0 0 47 0 0 48 0 0 49 0 0 50 0 0
51 0 0 52 0 0 53 0 0 54 0 0 55 0 0
56 0 0 57 0 0 58 0 0 59 0 0 60 0 0
61 0 0 62 0 0 63 0 0 64 0 0 65 0 0
66 0 0 67 0 0 68 0 0 69 0 0 70 0 0
71 0 0 72 0 0 73 0 0 74 0 0 75 0 0
76 0 0 77 0 0 78 0 0 79 0 0 80 0 0
81 0 0 82 0 0 83 0 0 84 0 0 85 0 0
86 0 0 87 0 0 88 0 0 89 0 0 90 0 0
91 0 0 92 0 0 93 0 0 94 0 0 95 0 0
96 0 0 97 0 0 98 0 0 99 0 0 100 0 0
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
33 34
35 36
37 38
39 40
41 42
43 44
45 46
47 48
49 50
51 52
53 54
55 56
57 58
59 60
61 62
63 64
65 66
mapp18.epx

mapp18.epx
41 0 0 42 0 0 43 0 0 44 0 0 45 0 0 46 0 0 47 0 0 48 0 0 49 0 0 50 0 0 51 0 0 52 0 0 53 0 0 54 0 0 55 0 0 56 0 0 57 0 0 58 0 0 59 0 0 60 0 0 61 0 0 62 0 0 63 0 0 64 0 0 65 0 0 66 0 0 67 0 0 68 0 0 69 0 0 70 0 0 71 0 0 72 0 0 73 0 0 74 0 0 75 0 0 76 0 0 77 0 0 78 0 0 79 0 0 80 0 0 81 0 0 82 0 0 83 0 0 84 0 0 85 0 0 86 0 0 87 0 0 88 0 0 89 0 0 90 0 0 91 0 0 92 0 0 93 0 0 94 0 0 95 0 0 96 0 0 97 0 0 98 0 0 99 0 0 100 0 0

mapp20.dgibi

opti echo 1;
opti dime 3 elen cub8;
opti trac pac ftra 'mapp20_mesh.ps';
opti sauv form 'mapp20.msh';

# Define mesh data

lhp3d = 12.0;
lhp1d = 4.0;
llp1d = 6.0;
llp3d = 10.0;
hnp3d = 1200;
hnp1d = 400;
nlp1d = 600;
nlp3d = 1000;
h = lhp3d / nhp3d;
tol = h / 10;
p0 = 0 0 0;
p1 = 0 h 0;
p2 = 0 b h;
p3 = 0 b 0;
bashp3d = manu qua4 p0 p1 p2 p3;
hp3d = bashp3d volu tran nhp3d (lhp3d 0 0);
baslp3d = bashp3d plus ((lhp3d+lhp1d+llp1d) 0 0);
lp3d = baslp3d volu tran nlp3d (llp3d 0 0);
p1d1 = p0 plus (lhp3d 0 0);
p1d2 = p1d1 plus (lhp1d 0 0);
p1d3 = p1d2 plus (llp1d 0 0);
hp1d = p1d1 d nhp1d p1d2;
lp1d = p1d2 d nlp1d p1d3;
hp = hp3d et hp1d;

# Define post-treatment

trac 6 7 8 9 106 107 108 109 AXES 1.0 'PRES [PA]';
list 6 7 8 9 106 107 108 109 AXES 1.0 'PRES [PA]';
trac 16 17 18 19 116 117 118 119 AXES 1.0 'DENS [KG/M3]';
list 16 17 18 19 116 117 118 119 AXES 1.0 'DENS [KG/M3]';
cold noir noir noir noir noir noir noir noir

post treatment

resu allic gao pcr
sort grap axe 1.0 't [s]'
scou 6 'p60' ecou comp 1 t 60.e-3 sae 1.0 'abc' lect 1 pas 1 101 term
scou 7 'p60' ecou comp 1 t 60.e-3 sae 1.0 'abc' lect 1 pas 1 101 term
scou 8 'p70' ecou comp 1 t 70.e-3 sae 1.0 'abc' lect 1 pas 1 101 term
scou 9 'p80' ecou comp 1 t 80.e-3 sae 1.0 'abc' lect 1 pas 1 101 term
rcou 106 'p50' fich 'mapp17.pun' rena 'p50_17'
rcou 107 'p60' fich 'mapp17.pun' rena 'p60_17'
rcou 108 'p70' fich 'mapp17.pun' rena 'p70_17'
rcou 109 'p80' fich 'mapp17.pun' rena 'p80_17'
trac 6 7 8 9 axes 1.0 'pres [pa]'
list 6 7 8 9 axes 1.0 'pres [pa]'
trac 16 17 18 19 axes 1.0 'dens [kg/m3]'
list 16 17 18 19 axes 1.0 'dens [kg/m3]'

# Define surface treatment

trac 6 7 8 9 106 107 108 109 axes 1.0 'pres [pa]'
list 6 7 8 9 106 107 108 109 axes 1.0 'pres [pa]'
cold noir noir noir noir noir noir noir noir noir

# Define geometrical data

mate gazp ro 13. pin 1.e6 gamm 1.402 pref 1.e5 cv 713.3
mate gazp ro 1.3 pin 1.e5 gamm 1.402 pref 1.e5 cv 713.3
init mapp form 'mapp11.map' matc obje lect tous term
ecri vfcu tfpe 10 e-3
fich alic tfre 10.e-3
!

# Define calculation parameters

time prog 50.e-3 term
opti note step 10 log 1
vfcu fode 6

diag droits lect hp term
turq lect 1p term

cou

mapp20.epx

MAPP20

ECOD

1CHBV win

CAST mesh

TMD EULE

DIME 20G 10 TERM ! Total n. of nodes in a TUBM junction

GEOM COUP hp3d lpd3 TOVF hlpd lplp facehp facecl

TURM rachp raclp TERM

COMP DIAM 001 0.011283792 LECT hlpd lplp TERM

RACC TURM LECT rachp TERM NTB LECT pld1 TERMINAL DTB 0.011283792

FACE LECT facehp TERM CCEF 1.0

RACC TURM LECT raclp TERM NTB LECT pld3 TERMINAL DTB 0.011283792

! Attention: the TURM elements (rachp and raclp) are NOT included
! in the "mesh" object (although they ARE indeed passed in from Cast3m).
! For this reason we must add them explicitly in the GROU directive below
! if we want to have them in the extracted element groups.

GROU 2 'nrachp' LECT mesh rachp raclp TERM

COOL RUGD LECT hp TERM

TURQ LECT hp TERM

VERT LECT hp TERM

MATE GAZP RO 1.20 GAMM 1.4 CV 720 PINI 1.0E5 PREF 1.0E5

MATE GAZP RO 1.2 GAMM 1.4 CV 720 PINI 1.0E5 PREF 1.0E5

PINF

! same characteristics as the materials used for hp and lp, respectively
! In order to obtain a printout at least of the 3D VFCCs I am obliged
! if we want to have them in the extracted element groups.

GROU 2 'nrachp' LECT mesh rachp raclp TERM

COMP DIAM 001 0.011283792 LECT hlpd lplp TERM

RACC TURM LECT raclp TERM

FACE LECT facehp TERM COEF 1.0

NTUB LECT p1d3 TERM DTUB 0.011283792

NTUB LECT p1d1 TERM DTUB 0.011283792

! Attention: the TUBM elements (rachp and raclp) are NOT included
! in the "mesh" object (although they ARE indeed passed in from Cast3m).
! For this reason we must add them explicitly in the GROU directive below
! if we want to have them in the extracted element groups.

GROU 2 'nrachp' LECT mesh rachp raclp TERM

COOL RUGD LECT hp TERM

TURQ LECT hp TERM

VERT LECT hp TERM

CDEF VMPC VRDF TDFR 1.E-3

! same characteristics as the materials used for hp and lp, respectively
! In order to obtain a printout at least of the 3D VFCCs I am obliged
! if we want to have them in the extracted element groups.

GROU 2 'nrachp' LECT mesh rachp raclp TERM

MATE GAZP RO 1.20 GAMM 1.4 CV 720 PINI 1.0E5 PREF 1.0E5

MATE GAZP RO 1.2 GAMM 1.4 CV 720 PINI 1.0E5 PREF 1.0E5

PINF

! same characteristics as the materials used for hp and lp, respectively
! In order to obtain a printout at least of the 3D VFCCs I am obliged
! if we want to have them in the extracted element groups.

GEOM CUVF hp3d lp3d TUVF hp1d lp1d CL3D facehp facelp

DIME JONC 10 TERM ! Total n. of nodes in a TUBM juncton

CAST mesh

CONV win

MAPP21

mapp21.epx

MAPP21

ECOD

1CHBV win

CAST mesh

TMD EULE

DIME 20G 10 TERM ! Total n. of nodes in a TUBM junction

GEOM COUP hp3d lpd3 TOVF hlpd lplp facehp facecl

TURM rachp raclp TERM

COMP DIAM 001 0.011283792 LECT hlpd lplp TERM

RACC TURM LECT rachp TERM NTB LECT pld1 TERMINAL DTB 0.011283792

FACE LECT facehp TERM CCEF 1.0

RACC TURM LECT raclp TERM NTB LECT pld3 TERMINAL DTB 0.011283792

! Attention: the TURM elements (rachp and raclp) are NOT included
! in the "mesh" object (although they ARE indeed passed in from Cast3m).
! For this reason we must add them explicitly in the GROU directive below
! if we want to have them in the extracted element groups.

GROU 2 'nrachp' LECT mesh rachp raclp TERM

COOL RUGD LECT hp TERM

TURQ LECT hp TERM

VERT LECT hp TERM

CDEF VMPC VRDF TDFR 1.E-3

! same characteristics as the materials used for hp and lp, respectively
! In order to obtain a printout at least of the 3D VFCCs I am obliged
! if we want to have them in the extracted element groups.

GROU 2 'nrachp' LECT mesh rachp raclp TERM

MATE GAZP RO 1.20 GAMM 1.4 CV 720 PINI 1.0E5 PREF 1.0E5

MATE GAZP RO 1.2 GAMM 1.4 CV 720 PINI 1.0E5 PREF 1.0E5

PINF

! same characteristics as the materials used for hp and lp, respectively
! In order to obtain a printout at least of the 3D VFCCs I am obliged
! if we want to have them in the extracted element groups.

GEOM CUVF hp3d lp3d TUVF hp1d lp1d CL3D facehp facelp

DIME JONC 10 TERM ! Total n. of nodes in a TUBM juncton

CAST mesh

CONV win

mapp21.dgibi

opti echo 1;
opti dime 3 elem cub8;
opti trac pac ftra 'mapp21_mesh.psa';
opti savf form 'mapp21.msh';

* lhp3d = 12.0;
lhp3d = 4.0;
lhp3d = 6.0;
lhp3d = 10.0;

* nraclp = 1000;

* mesh = flui et facehp et facelp et raclp et raclp;
* tass mesh noo;
* sauv form mesh;
* trac cach qual mesh;
* fin;

Post-treatment (space curves from alice file)

ECOD

OPTI PRIN

MESH ALCC 'mapp20.ali' GARD FGCH

COMP MGO 1 'xaco' LECT flui TERM

COND XL1 0 Y1 0 X2 0 Y2 0 Z0 0 T0L 0.0001

SOFT GRAP

PERF 'mapp20.psn'

AXE 1.0 'Time [s]'

SCOU 61 'p_21' NSTO 21 SAKE 1.0 'curr_abscissa' LECT xaco TERM

ECHO COMP 1

SCOU 62 'r_21' NSTO 21 SAKE 1.0 'curr_abscissa' LECT xaco TERM

ECHO COMP 2

SCOU 65 'v_21' NSTO 21 SAKE 1.0 'curr_abscissa' LECT xaco TERM

VCVI COMP 1

DCOU 71 'v_mna' SHTU GAMM 1.4 RMD 14.4 HDP 1.2 E2MT 2.08333E5

LEMN 16 LEPD 16

TIME 20.E-3 HRA3 30 VAR1

DCOU 72 'r_mna' SHTU GAMM 1.4 RMD 14.4 HDP 1.2 E2MT 2.08333E5

LEMN 16 LEPD 16

TIME 20.E-3 HRA3 30 VAR1

DCOU 75 'v_mna' SHTU GAMM 1.4 RMD 14.4 HDP 1.2 E2MT 2.08333E5

LEMN 16 LEPD 16

TIME 20.E-3 HRA3 30 VAR1

TRAC 61 71 AXES 1.0 'PRESS. [PA]'
mapp22.epx

Post-treatment (space curves from slice file)

ECNO ECGN VPCC TRTR 1.6-3
  NOPG NOEL
  FCH ALIC TRTR 1.6-3
  FCH FORM MAPP OBJE LECT bhp3 lhp3 bhp1 lhp1 TERM
  TIME PXRG 10.6-3 TERM
OPTI NOTE COTA 0.75
STEP 10 LOG 1
  VPCC FORM 6 'hlic solver
  DGB 2 ' order in space
  DTSP 2 ' order in time
  RECO 1 ' Not accepted by CAL_VPCC.1D
CALC TINI 0 TEND 20.6-3
FIN

mapp21p.epx

Post-treatment (space curves from slice file)

ECNO
OPTI PRIN
NDA LIST 65 AXES 1.0 'VELOC. [M/S]'
  COLO NOIR ROUG
  TRAC 65 75 AXES 1.0 'VELOC. [M/S]'
LIST 62 AXES 1.0 'DENS. [KG/M3]'
  COLO NOIR ROUG
  TRAC 62 72 AXES 1.0 'DENS. [KG/M3]'
LIST 61 AXES 1.0 'PRESS. [PA]'
  COLO NOIR ROUG
  TRAC 61 71 AXES 1.0 'PRESS. [PA]'
DCOU 75 'v_ana' SHTU GAMM 1.4 ROM 14.4 ROP 1.2 EINT 2.08333E5
DCOU 72 'r_ana' SHTU GAMM 1.4 ROM 14.4 ROP 1.2 EINT 2.08333E5
DCOU 71 'p_ana' SHTU GAMM 1.4 ROM 14.4 ROP 1.2 EINT 2.08333E5
SCOU 65 'vx_21' NSTO 21 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
SCOU 62 'ro_21' NSTO 21 SAXE 1.0 'curr_abscissa' LECT xaxo TERM

mapp22.dgibi

opti echo 1;
opti dime 3 elem cub;
opti trac psc ftra 'mapp22_mesh.ps';
opti sauv form 'mapp22.mah';
  lhdp = 12.0;
  lhpl = 4.0;
  lhlp = 6.0;
  lhlp = 10.0;
  shpl = 1200;
  shpl = 400;
  shpl = 600;
  shpl = 1000;
  h = lhdp / shhd;
tol = h / 10;
  p0 = 0.0;
p1 = h / 10;
p2 = 0.0;
  p3 = 0.0;
bashdp = manu qu4 p0 p1 p2 p3;
bashp = bashdp volu tran shdp (bashdp 0.0);
bashl = bashdp plus ((bashdp-<bashp+1bashdp> 0.0));
lhdp = bashl volu tran shlp (bashl 0.0);
lhpl = bashp plus (bashp 0.0);
lhlp = p0 plus (bashlp 0.0);
lhpl = p0 plus (bashlp 0.0);
lhdp = p0 plus (bashlp 0.0);
lhlp = p0 plus (bashlp 0.0);
  b = bashd et blp;
  lp = lhdp et lhp;
  flu = bp et b;
  * raccords 34-1d
  * facech = bashdp plus (1bashdp 0.0);
  pfacch = chan prin facech;
  alim topo (pfacch et bhp);
  faceb = bashlp;
  pfacch = chan prin facech;
  racch = maus pru (pfacch et facech);
  racch = maus pru (pfacch et facech);
  mesh = flui et facech et facech et racch et racch;
  tass mesh nopp;

mapp22.epx

Post-treatment (space curves from slice file)

ECNO
OPTI PRIN
NDA LIST 65 AXES 1.0 'VELOC. [M/S]'
  COLO NOIR ROUG
  TRAC 65 75 AXES 1.0 'VELOC. [M/S]'
LIST 62 AXES 1.0 'DENS. [KG/M3]'
  COLO NOIR ROUG
  TRAC 62 72 AXES 1.0 'DENS. [KG/M3]'
LIST 61 AXES 1.0 'PRESS. [PA]'
  COLO NOIR ROUG
  TRAC 61 71 AXES 1.0 'PRESS. [PA]'

mapp22p.epx

Post-treatment (space curves from slice file)

ECNO
OPTI PRIN
NDA LIST 65 AXES 1.0 'VELOC. [M/S]'
  COLO NOIR ROUG
  TRAC 65 75 AXES 1.0 'VELOC. [M/S]'
LIST 62 AXES 1.0 'DENS. [KG/M3]'
  COLO NOIR ROUG
  TRAC 62 72 AXES 1.0 'DENS. [KG/M3]'
LIST 61 AXES 1.0 'PRESS. [PA]'
  COLO NOIR ROUG
  TRAC 61 71 AXES 1.0 'PRESS. [PA]'

mapp22.epx
mapp30.dgibi

```
TRAC 62 162 AXES 1.0 'DENS. [KG/M3]'
COLO NOIR VERT
TRAC 65 165 AXES 1.0 'VELOC. [M/S]'
COLO NOIR VERT
FIN

mapp30.epx

```

mapp31.dgibi

```
TRAC 62 162 AXES 1.0 'DENS. [KG/M3]'
COLO NOIR VERT
TRAC 65 165 AXES 1.0 'VELOC. [M/S]'
COLO NOIR VERT
FIN

mapp31.epx

```

mapp32.dgibi

```
TRAC 62 162 AXES 1.0 'DENS. [KG/M3]'
COLO NOIR VERT
TRAC 65 165 AXES 1.0 'VELOC. [M/S]'
COLO NOIR VERT
FIN

mapp32.epx

```

mapp33.dgibi

```
TRAC 62 162 AXES 1.0 'DENS. [KG/M3]'
COLO NOIR VERT
TRAC 65 165 AXES 1.0 'VELOC. [M/S]'
COLO NOIR VERT
FIN

mapp33.epx

```
vega45.epx

-DEBPROC- psxtail cplat 'MAILLAGE' ch2*POINT' px='POINT' 
  vr='POINT' tol='FLOATANT' fac='FLOATANT';
sh2 = cplat poin cyli ch2 (ch2 plus px) tol; 
ho2 = cplat elem appu stri nho2; 
ho2 = ho2 poin i; i = i + 1; 
fin loop2; 
fimpan; 
-----------
-DEBPROC- nxshrink frameb*'MAILLAGE' cen*POINT'
  tol*FLOTTANT' fac*FLOTTANT';
ps = 1.0 0.0; 
rx = 1.25E-2; 
tol = 1.E-4; 
fac = 1.016; 
psxtail cplat ch2 ps tr vin fac tol; 
psxtail cplat ch2 ps tr vin fac tol; 
psxtail cplat ch2 ps tr vin fac tol; 
psxtail cplat ch2 ps tr vin fac tol; 
psxtail cplat ch2 ps tr vin fac tol; 
psxtail cplat ch2 ps tr vin fac tol; 
psxtail cplat ch2 ps tr vin fac tol; 
trac osil qual plate; 
oscil cplat qual iframeb; fac = 0.996; 
psxtail iframeb ch1 r face; 
psxtail iframeb ch1 r face; 
psxtail iframeb ch1 r face; 
psxtail iframeb ch1 r face; 
trac osil qual iframeb; 
opti sauv form 'vega45.mah'; 
taa8 mesh npp; 
fin form mesh; 
fin;
```
geom cub8 ecub8
dime adap npoi 100000
eos 1.0
def adap wpd1 100000
wpd1 200000
cl3d 200000
uq4e 200000
fin 200000
end 200000
jdoc 475 /* total n. of nodes in a TUBM junction
vole 1.0
term
seed cub8 wcb8
pre
```
123456 UPTO 30.0E-3 LECT plate TERM
INIT SKIP UPTO 3.0E-3 VFCC
ADAP IMAT TIME 3.0E-3
  2 MATE 2 OBJ LECT fluid TERM
INSI SURF LECT nen TERM
  MATE 3 OBJ LECT fluid TERM
  OUTS SURF LECT nen TERM
  INSI SURF LECT nen TERM
  MATE 4 OBJ LECT fluid TERM
  OUTS SURF LECT nen TERM
ECRI DEPL VITE ECRD FAIL TPRE 1.0E-3
  Poin LECT cen TERM
  ELEM LECT 1 TERM
FICN ALIT FREQ 0 TPRE 1.0
  TIME PRG 0 0.00 0.5D-3 28.0D-3 50.0D-3 30.0D-3 30.0D-3
  TIME PROG 0.0 1.00D-3 40.0D-3 30.0D-3
OPTI NOTE CSTA 0.4
  LOG 1
JADM LIMIT
  FANT 10x3 LECT nen TERM 1.0q4gs TERM
  PINS GRID DFIN 1.0
VFCC FOD 6 ! Alic solver
  GDR 2 ! order in space
STOP 2 ! order in time
RENO 1 ! Not accepted by CAL_VFCC_1D

vega53a.epx
Post-treatment
ECRD
CONV WIN
NEBU SPLI ALIC 'vega53.al! GARD PKC
COMD COUL NODG LECT nen TERM
VFCC FCON 6 ! hllc solver
KNOR UPTO 30.0E-3 LECT plate lframeb uframe TERM
FLS CUBS 2 ! For the inverse mapping
QUAD STAT 0.71 0.5D-3
CALC TINI 0 TEND 10.0E-3
FIN

vega53b.epx
POST
ECRD
NEBU SPLI ALIC 'vega53.al! GARD PKC
COMD COUL NODG LECT nen TERM
VFCC FCON 6 ! hllc solver

vega53c.epx
POST
ECRD
NEBU SPLI ALIC 'vega53.al! GARD PKC
**vega53l.epx**

**ECHO**

**CALCUL**

**Post-treatment**

**vuga53k.epx**

**ECHO**

**CALCUL**

**Post-treatment**

**vuga53m.epx**

**ECHO**

**CALCUL**

**Post-treatment**
Post-treatment 

ECRD

CS DNV

MATE SPLIT ALIC 'vega61.ali' GARD PSCRI

SORT GRAP

AXTE 1.0 'Time [s]' 

CONV 1 'SENS1' ECRD COMP 1 LECT S1 TERM

CONR 2 'SENS2' ECRD COMP 1 LECT S2 TERM 

TRA C 2 AXES 1.0 'Pres [Pa]' \ YERZ 

COLD BLU E 

LIST 1 2 AXES 1.0 'Pres [Pa]' \ YERZ 

LIST 10 AXES 1.0 'Disp [m]' \ YERZ 

FIN

vega61b.epx 

vega61c.epx 

vega61k.epx
vega61m.epx

Post-treatment
ECOO

CONV WIN
RESU SPLI ALIC "vega61.ali" GARD PSCR
COMP COOL RUGD LECT nent. TERM
VEKT LECT nent. TERM
TURQ LECT nent. TERM
SORT VISU NFDI 1
PLAY
CAME 1 EYE -1.72476E+01 0.00000E+00 -1.87219E-10
Q 7.07107E-01 0.00000E+00 -7.07107E-01 0.00000E+00
VIEW 1.00000E+00 0.00000E+00 2.05103E-01
UP 0.00000E+00 0.00000E+00 1.00000E+00
POV 2.48819E+01
NAVIGATION MODE: ROTATING CAMERA
CENTER : -1.63580E-01 0.00000E+00 0.00000E+00
RSPHERE : 2.34052E-01
RADIUS : 9.12804E-01
ASPECT : 1.00000E+00
REAR : 6.78726E-01
FAR : 1.38001E+00
SCEN GEOM NAVI FREE
LINE HEDU
FACE SSBC
LIMA ON
Slero CAMI 1 NFDI 1
TRAC OFFS FICH BMP OBJE LECT nent. TERM NFAI SYXY SYXZ
AMPD 0 KEND
FREQ 1
GOTH LOOP 6 OFFS FICH BMP OBJE LECT nent. TERM NFAI SYXY SYXZ
AMPD 0 KEND
ENDPLAY
FIN

vega61n.epx

Post-treatment
ECOO

CONV WIN
RESU SPLI ALIC "vega61.ali" GARD PSCR
COMP COOL RUGD LECT nent. TERM
VEKT LECT nent. TERM
TURQ LECT nent. TERM
SORT VISU NFDI 1
PLAY
CAME 1 EYE -1.72476E+01 0.00000E+00 -1.87219E-10
Q 7.07107E-01 0.00000E+00 -7.07107E-01 0.00000E+00
VIEW 1.00000E+00 0.00000E+00 2.05103E-01
UP 0.00000E+00 0.00000E+00 1.00000E+00
POV 2.48819E+01
NAVIGATION MODE: ROTATING CAMERA
CENTER : -1.63580E-01 0.00000E+00 0.00000E+00
RSPHERE : 2.34052E-01
RADIUS : 9.12804E-01
ASPECT : 1.00000E+00
REAR : 6.78726E-01
FAR : 1.38001E+00
SCEN GEOM NAVI FREE
LINE HEDU
FACE SSBC
LIMA ON
Slero CAMI 1 NFDI 1
TRAC OFFS FICH BMP OBJE LECT nent. TERM NFAI SYXY SYXZ
AMPD 0 KEND
FREQ 1
GOTH LOOP 6 OFFS FICH BMP OBJE LECT nent. TERM NFAI SYXY SYXZ
AMPD 0 KEND
ENDPLAY
FIN

vega61o.epx

Post-treatment
ECOO

CONV WIN
RESU SPLI ALIC "vega61.ali" GARD PSCR
COMP COOL RUGD LECT nent. TERM
VEKT LECT nent. TERM
TURQ LECT nent. TERM
SORT VISU NFDI 1
PLAY
CAME 1 EYE -1.72476E+01 0.00000E+00 -1.87219E-10
Q 7.07107E-01 0.00000E+00 -7.07107E-01 0.00000E+00
VIEW 1.00000E+00 0.00000E+00 2.05103E-01
UP 0.00000E+00 0.00000E+00 1.00000E+00
POV 2.48819E+01
NAVIGATION MODE: ROTATING CAMERA
CENTER : -1.63580E-01 0.00000E+00 0.00000E+00
RSPHERE : 2.34052E-01
RADIUS : 9.12804E-01
ASPECT : 1.00000E+00
REAR : 6.78726E-01
FAR : 1.38001E+00
SCEN GEOM NAVI FREE
LINE HEDU
FACE SSBC
LIMA ON
Slero CAMI 1 NFDI 1
TRAC OFFS FICH BMP OBJE LECT nent. TERM NFAI SYXY SYXZ
AMPD 0 KEND
FREQ 1
GOTH LOOP 6 OFFS FICH BMP OBJE LECT nent. TERM NFAI SYXY SYXZ
AMPD 0 KEND
ENDPLAY
FIN
"DEBPROC" pascal cplat=\'MAILLAGE\' ch2=\'POINT\' px=\'POINT\';
vr=\'POINT\'; tol=\'FLOTTANT\'; fac=\'FLOTTANT\';
shb2 = cplat poin cyli ch2 (ch2 plus px) (ch2 plus vr) tol;
ho2 = cplat elem appu stri shb2;
hho2 = ho2 home fac ch2;
i = 0;
repe loop2 (shb2 ho2);
i = i + 1;
ni = ho2 poin i;
fin loop2;
finproc;

emoi2 cach qual lframeb;
oeil2 = 100000 0 -30000;
pxswell cplat ch2 px vr tol fac;
vr = 0 r 0;
px = 1 0 0;

ch2 = 0 0 26.E-2;
ch1 tour 75.0 p0 p0p;
ch5 = ch1 tour 60.0 p0 p0p;
ch4 = ch1 tour 45.0 p0 p0p;
ch3 = ch1 tour 30.0 p0 p0p;
ch2 = ch1 tour 15.0 p0 p0p;
p0p = 1 0 0;
p0 = 0 0 0;
ch1 = 0 26.E-2 0;
cplat = cont plate;
oeil = 100000 0 0;

opti trac psc ftra 'vega62_mesh.ps';
* 

PARO PSIL 0.02
GAZP RO 1.189 GAMM 1.4 CV 719.286 PINI 1.011E5 PREF 1.011E5
GAZP RO 1.202 GAMM 1.4 CV 719.286 PINI 1.011E5 PREF 1.011E5
GAZP RO 16.307 GAMM 1.4 CV 719.286 PINI 13.723E5 PREF 1.011E5
GAZP RO 31.412 GAMM 1.4 CV 719.286 PINI 26.435E5 PREF 1.011E5
GAZP RO 46.517 GAMM 1.4 CV 719.286 PINI 39.147E5 PREF 1.011E5

opti echo 1;
opti rest form \textquoteleft{}vega62.mah\textquoteleft{};
rest form;
list;
* 
oubi mesh;
oubi psc;
oubi px;
oubi px2;
oubi px3;
oubi px4;
oubi px5;
oubi px6;
oubi px7;
oubi px8;
oubi px9;
oubi px10;
oubi px11;
oubi px12;
oubi px13;
oubi px14;
oubi px15;
oubi px16;
oubi px17;
oubi px18;
oubi px19;
oubi px20;
* 
list;

scen = spec et devi et stubd et radcdd et pfasecd et fascd et
fluid et bout et fond;

print mesh;
* 

scen = spec et devi et stubd et radcdd et pfasecd et fascd et
fluid et bout et fond;

print mesh;
*
Post-treatment

ECHO

VEGA62C.epx

Post-treatment

ECHO

VEGA62L.epx

SUPPLE ALIC TEMP 'vega62.ali' GARD PSCR

SORT GRAP

RESU ALIC TEMP 'vega62.alt' GARD PSCR

ECHO

vega62c.epx

vuga622.epx
vega63.epx

Post-treatment

ECHO

CONV WIN

RESU SPLI ALIC 'vega63_all' GAND PSOR

COMP COUL ROUG LECT mems TERM

RESU SPLI ALIC 'vega63.epx' FICH ALIC PSCR

ECHO

Post-treatment

ECHO

CONV WIN
## List of input files

<table>
<thead>
<tr>
<th>M</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>mapp00.epx</td>
<td>vega52c.epx</td>
</tr>
<tr>
<td>mapp01.epx</td>
<td>vega52l.epx</td>
</tr>
<tr>
<td>mapp02.epx</td>
<td>vega52l.epx</td>
</tr>
<tr>
<td>mapp03.epx</td>
<td>vega53.epx</td>
</tr>
<tr>
<td>mapp04.epx</td>
<td>vega53.epx</td>
</tr>
<tr>
<td>mapp05.epx</td>
<td>vega53a.epx</td>
</tr>
<tr>
<td>mapp06.epx</td>
<td>vega53b.epx</td>
</tr>
<tr>
<td>mapp07.epx</td>
<td>vega53b.epx</td>
</tr>
<tr>
<td>mapp08.epx</td>
<td>vega53c.epx</td>
</tr>
<tr>
<td>mapp09.epx</td>
<td>vega53c.epx</td>
</tr>
<tr>
<td>mapp09b.epx</td>
<td>vega53k.epx</td>
</tr>
<tr>
<td>mapp10.epx</td>
<td>vega53k.epx</td>
</tr>
<tr>
<td>mapp11.epx</td>
<td>vega53l.epx</td>
</tr>
<tr>
<td>mapp12.epx</td>
<td>vega53l.epx</td>
</tr>
<tr>
<td>mapp16.epx</td>
<td>vega53m.epx</td>
</tr>
<tr>
<td>mapp17.epx</td>
<td>vega53m.epx</td>
</tr>
<tr>
<td>mapp18.epx</td>
<td>vega53n.epx</td>
</tr>
<tr>
<td>mapp20.epx</td>
<td>vega53n.epx</td>
</tr>
<tr>
<td>mapp20p.epx</td>
<td>vega53o.epx</td>
</tr>
<tr>
<td>mapp21.dgibi</td>
<td>vega53o.epx</td>
</tr>
<tr>
<td>mapp21.epx</td>
<td>vega61.dgibi</td>
</tr>
<tr>
<td>mapp21p.epx</td>
<td>vega61.epx</td>
</tr>
<tr>
<td>mapp22.dgibi</td>
<td>vega61.epx</td>
</tr>
<tr>
<td>mapp22.epx</td>
<td>vega61a.epx</td>
</tr>
<tr>
<td>mapp22p.epx</td>
<td>vega61a.epx</td>
</tr>
<tr>
<td>mapp30.dgibi</td>
<td>vega61b.epx</td>
</tr>
<tr>
<td>mapp30.epx</td>
<td>vega61b.epx</td>
</tr>
<tr>
<td>mapp31.dgibi</td>
<td>vega61c.epx</td>
</tr>
<tr>
<td>mapp31.epx</td>
<td>vega61c.epx</td>
</tr>
<tr>
<td>mapp31.epx</td>
<td>vega61k.epx</td>
</tr>
<tr>
<td>mapp32.dgibi</td>
<td>vega61k.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega61l.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega61l.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega62a.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega62a.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega62b.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega62b.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega62c.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega62c.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega63.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega63.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega63a.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega63a.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega63b.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega63b.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega63c.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega63c.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega63k.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega63k.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega63l.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega63l.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega63m.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega63m.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega63n.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega63n.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega63o.epx</td>
</tr>
<tr>
<td>mapp32.epx</td>
<td>vega63o.epx</td>
</tr>
</tbody>
</table>
GETTING IN TOUCH WITH THE EU

In person
All over the European Union there are hundreds of Europe Direct information centres. You can find the address of the centre nearest you at: http://europea.eu/contact

On the phone or by email
Europe Direct is a service that answers your questions about the European Union. You can contact this service:
- by freephone: 00 800 6 7 8 9 10 11 (certain operators may charge for these calls),
- at the following standard number: +32 22999696, or
- by electronic mail via: http://europa.eu/contact

FINDING INFORMATION ABOUT THE EU

Online
Information about the European Union in all the official languages of the EU is available on the Europa website at: http://europa.eu

EU publications
You can download or order free and priced EU publications from EU Bookshop at: http://bookshop.europa.eu. Multiple copies of free publications may be obtained by contacting Europe Direct or your local information centre (see http://europa.eu/contact).
JRC Mission

As the science and knowledge service of the European Commission, the Joint Research Centre’s mission is to support EU policies with independent evidence throughout the whole policy cycle.