

STKO

A revolutionary toolkit for opensees



Massimo Petracca
Francesca Candeloro
Guido Camata
ASDEA Soft

massimo.petracca@unich.it m.petracca@asdea.net
f.candeloro@asdea.net
g.camata@unich.it
info@asdeasoft.net



Outline

- ◉ Introduction and motivations
- ◉ STKO: pre and post processor
- ◉ Adopted database: HDF5
- ◉ Proposed file format: MPCO
- ◉ New recorder class: MPCORecorder
- ◉ Numerical applications
- ◉ Conclusions and future works

1

Introduction and motivations

Introduction and motivations



Introduction and motivations

- Simulation of real- life structures
 - Real- life complex models
 - High- volume output data
 - Heterogeneous results
- Efficient layout of complex data in the database
- Lack of visualization tools for results on fibers and in general ID element
- Interaction with the database via scripting
 - Extrapolation
 - Manipulation
 - New custom results

2

STKO: pre and post processor

A brief description of the STKO pre and post processor



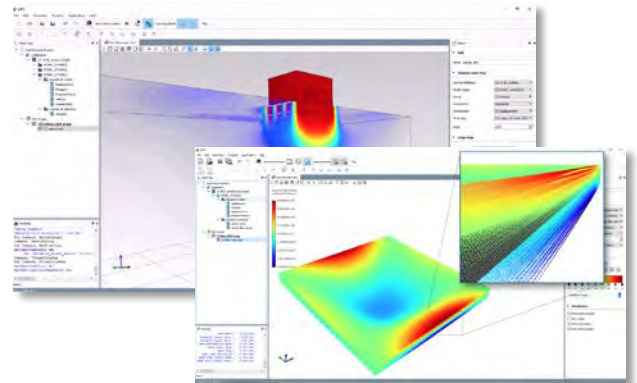
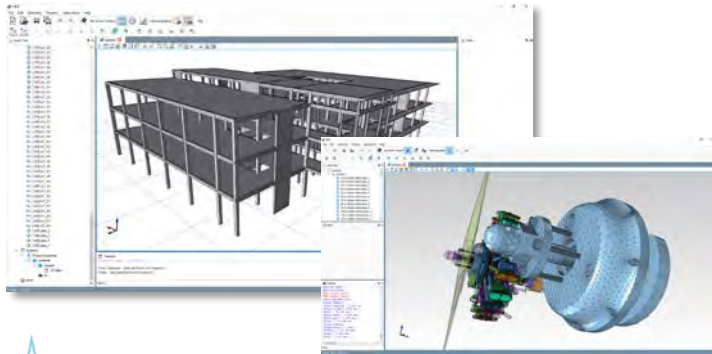
STKO: pre and post processor

Pre-processor

- CAD importer, modeler and mesher
- Based on OpenCascade library (<https://www.opencascade.com/>)
- Scripting interface for linking with the external solver

Post-processor

- HDF5- based output database (<https://www.hdfgroup.org/>)
- Standard plot tools
- Advanced plot tools for beam elements
- Scripting interface for interaction with the Database



3

Adopted database: HDF5

What is HDF5 and why did we choose it ?

For more info:

The HDF Group, "Hierarchical Data Format, version 5," 1997-2017. [Online]. Available: <http://www.hdfgroup.org/HDF5/>



Adopted database: HDF5

Why HDF5 ?

- **Opensource** (BSD license) library
- Completely **portable** file format
- **No limit** on the number or size of data objects in the collection
- Runs on a range of computational platforms, from laptops to massively parallel systems
- Several **languages**: C, C++, Fortran90, Java, Python
- **Parallel** I/O



Adopted database: HDF5

What does a HDF5 file look like?

- Hierarchical structure
- Groups (directories)
- Datasets (data)
- Attributes (metadata)

The screenshot displays the HDFView 2.13 application window. The left pane shows a hierarchical tree structure of the file, including groups like INFO, MODEL_STAGE, MODEL, NODES, ELEMENTS, SECTION_ASSIGNMENTS, and RESULTS. The right pane shows a table view of data from the 'STEP_201' dataset. The table has 21 rows and 6 columns, with values ranging from -5.4 to 1.03. The bottom status bar shows file information: 'STEP_201 (218950440, 2)', '8-bit floating-point, 24 x 1150', 'Number of attributes = 2', 'STEP = 201', and 'TIME = 3.8400000000000003'. A 'Log Info' and 'Metadata' button is visible at the bottom left of the window.

	0	1	2	3	4	5
0	-5.4772928	-5.4705916	-5.4639009	-5.4572099	-5.4505189	-5.4
1	8.0711872	8.0633663	8.0555255	8.0476946	8.0398837	8.0
2	5.6043845	5.5979854	5.5915863	5.5851872	5.5787881	5.5
3	-8.5598144	-7.1892133	-6.8306122	-4.4760111	-3.1134101	-1.7
4	-2.5520242	-2.5318599	-2.5116956	-2.5915312	-2.5713669	-2.5
5	1.9360548	1.7905239	1.6459832	1.6014434	1.3658017	1.2
6	1.9221287	1.9044548	1.8867831	1.8691113	1.8514394	1.8
7	-4.4901296	-4.4738241	-4.4575187	-4.4292432	-4.4069477	-4.3
8	-2.0304219	-2.0120445	-1.9952671	-1.9776897	-1.9601123	-1.9
9	2.0160200	2.0267360	2.0324560	2.0391770	2.0458960	2.05
10	5.7530334	5.6819958	5.6100583	5.5381207	5.4661832	5.3
11	1.8471195	1.8523627	1.8596459	1.8659089	1.8721720	1.87
12	-6.3295014	3.4342015	1.3197904	2.2951607	3.2725310	4.24
13	-7.7802725	-6.8342562	-5.8602436	-4.9422290	-3.9651443	-3.0
14	1.1135727	1.0178919	9.2203108	8.2628024	7.3048938	6.3
15	7.8050091	6.7057017	5.6064004	4.5070991	3.4077977	2.30
16	6.9482259	5.8439094	4.7415930	3.6392785	2.5369600	1.43
17	6.6136152	5.5118865	4.4101778	3.3084591	2.2067404	1.10
18	7.5915028	6.5151716	5.4388405	4.3625093	3.2861781	2.20
19	8.7289397	5.6501576	4.5713756	3.4925935	2.4138114	1.33
20	6.3985232	5.3197225	4.2409419	3.1621513	2.0833607	1.06

4

Proposed file format: MPCO

How to lay out complex data



Proposed file format: MPCO

What do we need to store in our output file?

- Model (+ model stages)

 - Nodes

 - Elements

 - Geometries + standard and custom integration rules

 - Sections and materials + assignments

- Results

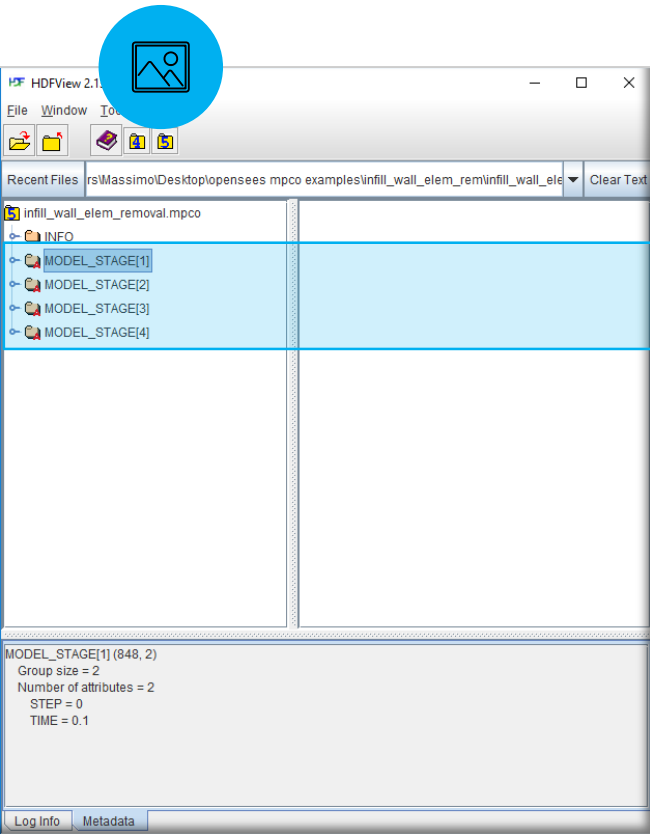
 - Results stored on nodes

 - Results stored on elements

 - On element nodes

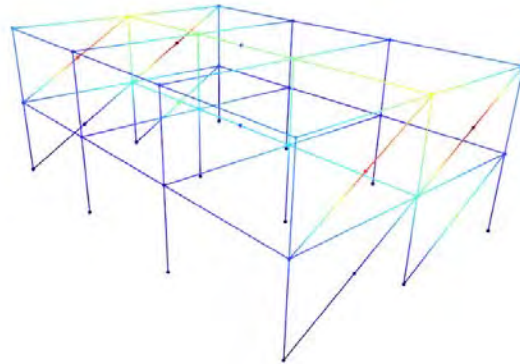
 - On integration points

 - On sub-integration points (fibers)



Model Stages:

- Changes in the original model
- Added/ removed elements/ nodes

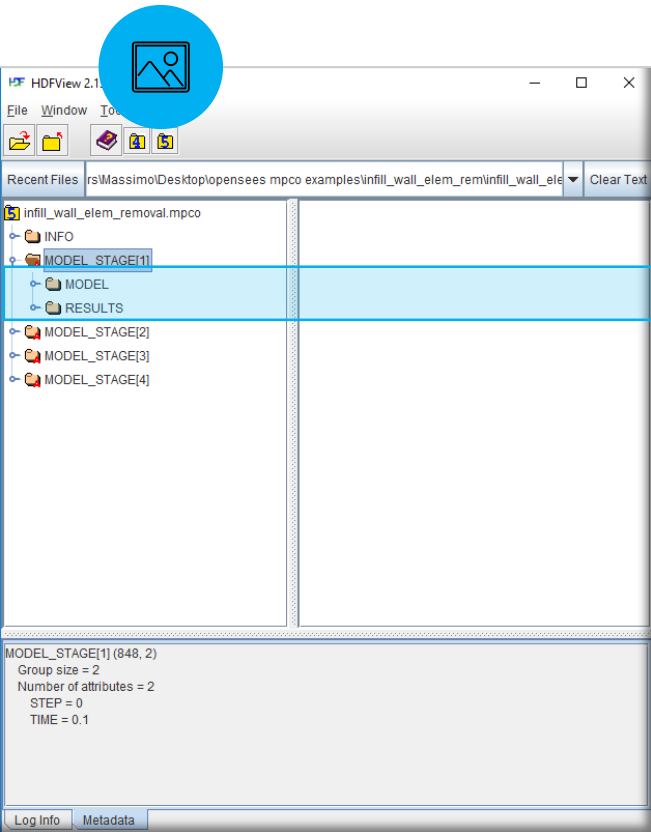


From OpenSees Structural Examples

Infill Wall Model and Element Removal

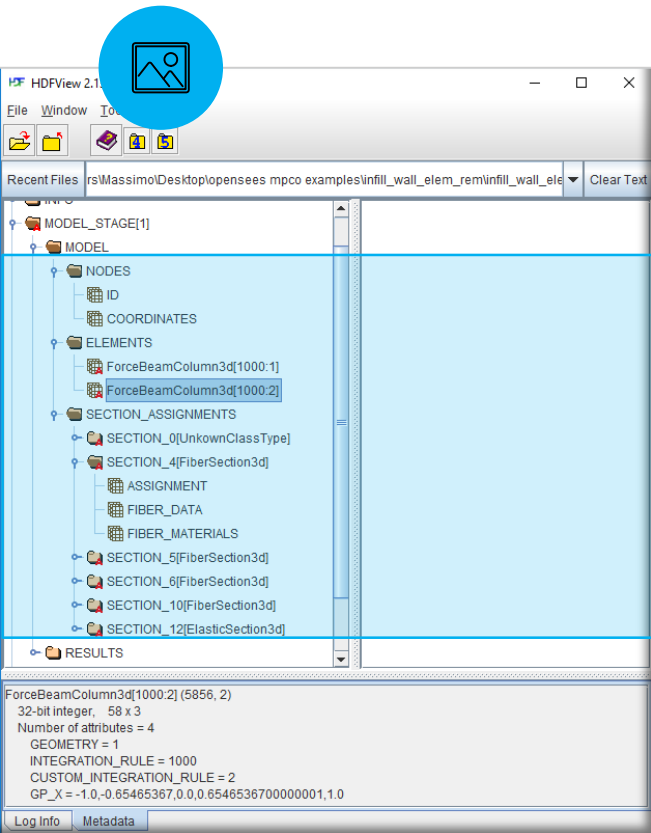
M. Selim Gunay and Khalid M. Mosalam, University of California, Berkeley

http://opensees.berkeley.edu/wiki/index.php/Infill_Wall_Model_and_Element_Removal



Each model stage group contains:

- Model informations
- Results



Each model group contains:

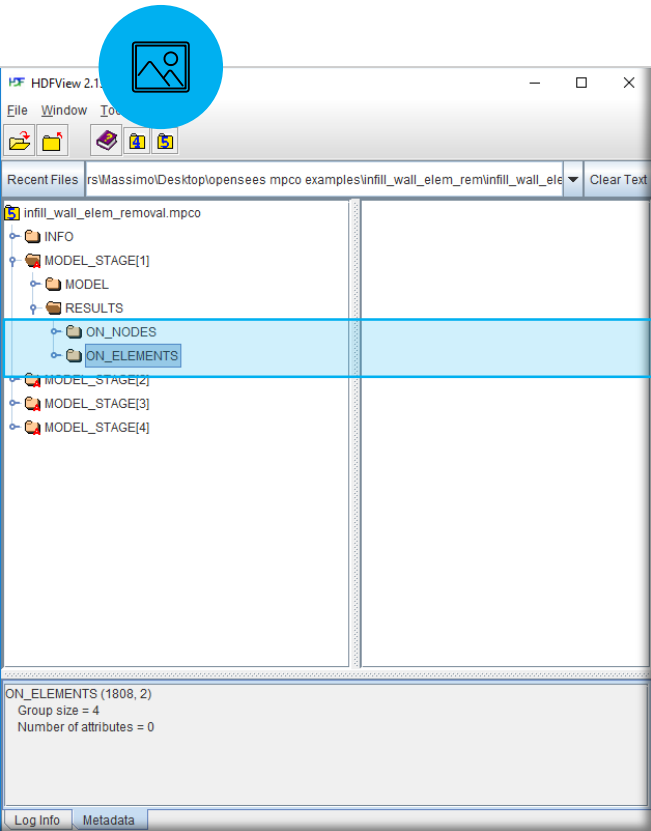
- Nodes
- Elements
 - Geometry
 - Standard and/or custom integration rules
- Section assignments
 - Element and gauss assignments
 - Fiber data
 - Fiber materials



Gauss points



Fibers



The result group contains:

- Results on nodes
- Results on elements



The screenshot shows the HDFView 2.1.1 interface. The left pane displays a tree view of the model results, with the following structure:

- MODEL_STAGE[1]
 - MODEL
 - RESULTS
 - ON_NODES
 - DISPLACEMENT
 - ID
 - DATA
 - STEP_0 (selected)
 - STEP_1
 - STEP_2
 - STEP_3
 - STEP_4
 - STEP_5
 - STEP_6
 - STEP_7
 - STEP_8
 - STEP_9

The right pane shows a table titled "STEP_0 at /MODEL_STAGE[1]/RESULTS/O...". The table has 18 rows and 4 columns. The first row is the header, and the subsequent rows contain numerical data.

	0	1	2
0	1.8671083...	0.0	5.4173411...
1	2.5210697...	0.0	-7.9410860...
2	3.7301289...	3.1775110...	-4.8182710...
3	-1.1686884...	-1.0418469...	5.6287281...
4	1.3074520...	1.1833085...	-9.5979745...
5	-1.6205916...	6.2258486...	-1.1316994...
6	1.7969975...	-7.9081071...	-4.1165042...
7	-1.2062959...	4.1885022...	-2.4922141...
8	-2.2105624...	5.6435280...	-1.9861882...
9	1.6797219...	-5.2924704...	2.1388886...
10	-5.8714879...	-4.8336435...	-1.4157904...
11	-5.1708992...	-3.2486966...	9.6039075...
12	8.7610252...	1.0645426...	7.4673306...
13	-6.0853085...	-8.3795606...	-1.7811022...
14	-5.9481299...	-7.6127927...	5.6661413...
15	4.6554248...	-7.7526975...	5.6662913...
16	-5.9479531...	-7.6130956...	-5.6661412...
17	4.6554211...	-7.7526912...	-5.6662912...

The bottom pane shows the metadata for the selected node:

STEP_0 (37256, 2)
64-bit floating-point, 46 x 3
Number of attributes = 2
STEP = 0
TIME = 0.1

The node result contains:

- 1 Dataset with selected node IDs
- 1 Data group
- 1 Dataset for each time step

HDFView 2.1.1

File Window Tools

Recent Files: rs\Massimo\Desktop\pensees mpco examples\infill_wall_elem_rem\infill_wall_ele... Clear Text

Tree View:

- ON_NODES
 - ON_ELEMENTS
 - section.force
 - section.deformation
 - section.fiber.stress
 - section.fiber.strain
 - ForceBeamColumn3d[1000:1:0]
 - ForceBeamColumn3d[1000:2:1]
 - ForceBeamColumn3d[1000:2:0]
 - META
 - MULTIPLICITY
 - GAUSS_IDS
 - NUM_COMPONENTS
 - COMPONENTS
 - ID
 - DATA
 - STEP_0
 - STEP_1
 - STEP_2

Table: STEP_0 at /MODEL_STAGE[1]/RESU... 0-based

	0	1	2
0	-1.7736773...	-1.7736773...	-1.7736773...
1	-4.1677783...	-4.1677783...	-4.1677783...
2	-1.7759530...	-1.7759530...	-1.7759530...
3	-9.9873919...	-9.9873919...	-9.9873919...
4	-3.4434772...	-3.4434772...	-3.4434772...
5	-1.0060644...	-1.0060644...	-1.0060644...
6	-9.9873919...	-9.9873919...	-9.9873919...
7	-3.4434772...	-3.4434772...	-3.4434772...
8	-1.0060644...	-1.0060644...	-1.0060644...
9	-1.7736773...	-1.7736773...	-1.7736773...
10	-4.1677783...	-4.1677783...	-4.1677783...
11	-1.7759530...	-1.7759530...	-1.7759530...
12	2.0050148...	2.3340434...	2.6630
13	-3.9754786...	-3.6465133...	-3.3177
14	-7.3952879...	-7.0662625...	-6.7377
15	4.4484592...	4.4038121...	4.3591
16	-1.5476067...	-1.5922959...	-1.6368
17	-4.9420097...	-4.9866599...	-5.0311

STEP_0 (935824, 2)
64-bit floating-point, 24 x 1150
Number of attributes = 2
STEP = 0
TIME = 0.1

Log Info Metadata

The element result contains:

- Groups based on:
 - Element type
 - Size of connectivity
 - Type of integration rule
 - Type of cross sections
- Each group contains:
 - Metadata group with:
 - Number of components + labels
 - Node or gauss ID
 - Multiplicity
 - 1 Dataset with selected element IDs
 - 1 Data group
 - 1 Dataset for each time step



PDF HDFView 2.1.1

File Window Tools

Recent Files: rts\Massimo\Desktop\opensees mpco examples\infill_wall_elem_rem\infill_wall_ele... Clear Text

ON_NODES
ON_ELEMENTS
section.force
section.deformation
section.fiber.stress
section.fiber.strain

ForceBeamColumn3d[1000:1:0]
ForceBeamColumn3d[1000:2:1]
ForceBeamColumn3d[1000:2:0]

META
MULTIPLICITY
GAUSS_IDS
NUM_COMPONENTS
COMPONENTS
ID
DATA
STEP_0
STEP_1
STEP_2

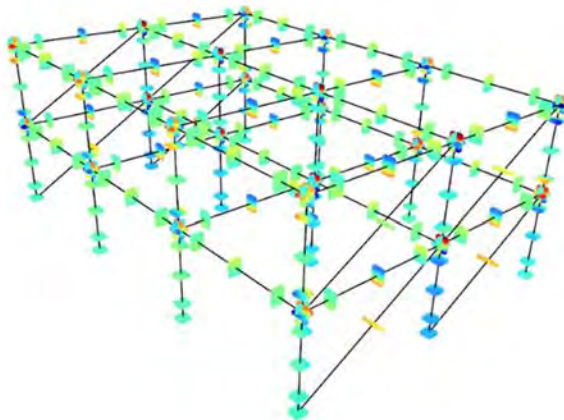
STEP_0 at /MODEL_STAGE[1]/RESU...
Table
0-based

	0	1	2
0	-1.7736773...	-1.7736773...	-1.7736773...
1	-4.1677783...	-4.1677783...	-4.1677783...
2	-1.7759530...	-1.7759530...	-1.7759530...
3	-9.9873919...	-9.9873919...	-9.9873919...
4	-3.4434772...	-3.4434772...	-3.4434772...
5	-1.0060644...	-1.0060644...	-1.0060644...
6	-9.9873919...	-9.9873919...	-9.9873919...
7	-3.4434772...	-3.4434772...	-3.4434772...
8	-1.0060644...	-1.0060644...	-1.0060644...
9	-1.7736773...	-1.7736773...	-1.7736773...
10	-4.1677783...	-4.1677783...	-4.1677783...
11	-1.7759530...	-1.7759530...	-1.7759530...
12	2.0050148...	2.3340434...	2.6630148...
13	-3.9754786...	-3.6465133...	-3.3174786...
14	-7.3952879...	-7.0662625...	-6.7372879...
15	4.4484592...	4.4038121...	4.3591592...
16	-1.5476067...	-1.5922959...	-1.6368067...
17	-4.9420097...	-4.9866599...	-5.0312097...

STEP_0 (935824, 2)
64-bit floating-point, 24 x 1150
Number of attributes = 2
STEP = 0
TIME = 0.1

Log Info Metadata

With the proposed layout we can obtain a large variety of plots:



5

New recorder class: MPCORecorder

Implementation of the HDF5 based recorder in OpenSees



New recorder class: MPCORRecorder

What element results are supported?

Everything, but the element MUST use the OPS_Stream!

MPCORRecorder heavily relies on OPS_Stream

```
MPCORRecorder.cpp
MPCORRecorder_ElementOutputDescriptorStream
struct MPCORRecorder_ElementOutputDescriptorHeader { ... };
#ifdef MPCOR_USE_UNORDERED_MAP
#endif
class MPCORRecorder_ElementOutputDescriptor { ... };
class MPCORRecorder_ElementOutputDescriptorStream : public OPS_Stream
{
public:
    MPCORRecorder_ElementOutputDescriptorStream(MPCORRecorder_ElementOutputDescriptor * _d
        : OPS_Stream(OPS_STREAM_TAGS_MPCORRecorder_ElementOutputDescriptorStream)
        , descr(_d)
        , current_level(0)
        , pending_close_tag(false)
    {}
    ~MPCORRecorder_ElementOutputDescriptorStream() {}
};
```

```
Response*
ForceBeamColumn3d::setResponse(const char **argv, int argc, OPS_Stream &output)
{
    Response *theResponse = 0;

    output.tag("ElementOutput");
    output.attr("elementType", "ForceBeamColumn3d");
    output.attr("eleTag", this->getTag());
    output.attr("node1", connectedExternalNodes[0]);
    output.attr("node2", connectedExternalNodes[1]);
}
```

```
for (int i=0; i<numSections; i++) {

    output.tag("GaussPointOutput");
    output.attr("number", i+1);
    output.attr("eta", xi[i]*L);

    Response *theSectionResponse = sections[i]->setResponse(&argv[1], argc-1, output);
}
```

6

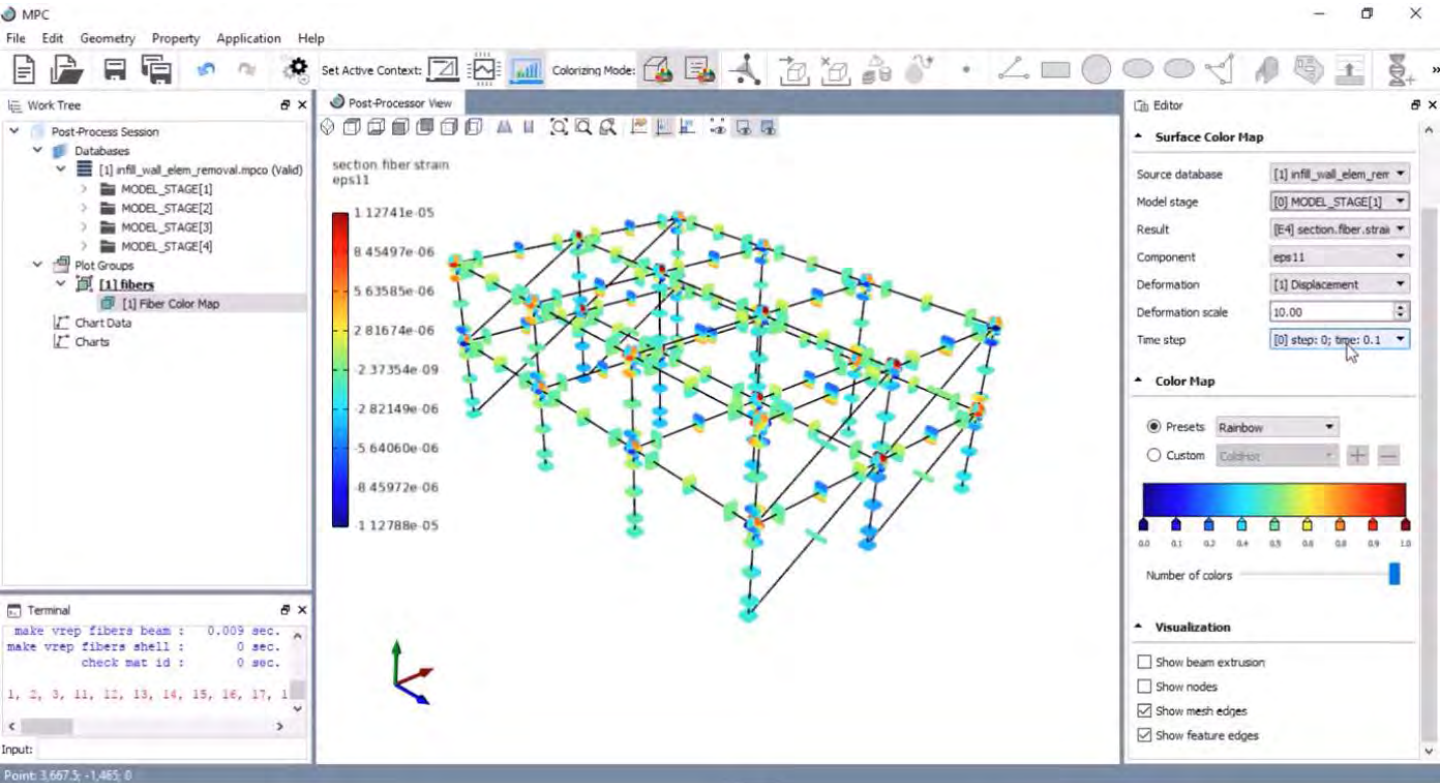
Numerical applications

Some examples using STKO for post-processing

Example of Soil-Foundation-Structure-Interaction

Visualization of frame elements, gauss plot and fiber plot

Visualization of shell elements, gauss plot and fiber plot



Element removal, fiber plot, chart data extraction

From OpenSees Structural Examples

Infill Wall Model and Element Removal

M. Selim Gunay and Khalid M. Mosalam, University of California, Berkeley

http://opensees.berkeley.edu/wiki/index.php/Infill_Wall_Model_and_Element_Removal



Conclusions

- HDF5 hierarchical structure is well suited for storing complex data
- A proper layout of complex data allows for advanced visualization tools such as fiber-section plot
- Future implementations
 - Parallel IO: now MPCORecorder works with OpenSees and OpenSeesMP (via partitioned databases), not with OpenSeesSP
 - Get local axes info from OpenSees



Thanks!

Any questions ?

Massimo Petracca
Francesca Candeloro
Guido Camata
ASDEA Soft

massimo.petracca@unich.it m.petracca@asdea.net
f.candeloro@asdea.net
g.camata@unich.it
info@asdeasoft.net



Credits

Special thanks to all the people who made and released these awesome resources for free:

- ◉ Presentation template by [SlidesCarnival](#)
- ◉ Photographs by [Unsplash](#)