Improvements to post–processing in Nektar++

Douglas Serson

Department of Mechanical Engineering
University of São Paulo

Nektar++ Workshop, June 2017
London

Introduction

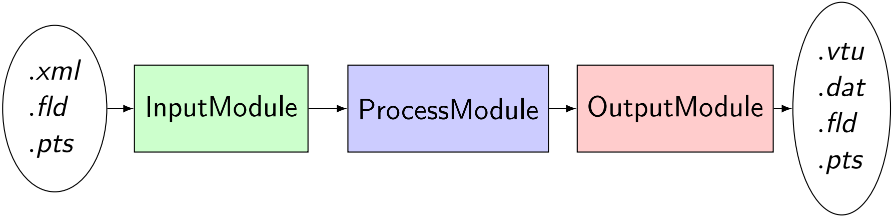
- I will present some new developments to FieldConvert
- At the moment, these changes are in the branch feature/FC-refactor
- Comments or suggestions are welcome

Introduction: FieldConvert

- Post-processing in Nektar++ is done using FieldConvert, e.g.

```
FieldConvert in1.xml in2.fld -m vorticity out.vtu
```

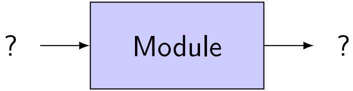
- FieldConvert is organized in modules, which are classified as Input, Process and Output



- It is possible to use many input and process modules in a single FieldConvert call

Problem

- FieldConvert modules operate on a Field object m_f, which contains: m_data/m_fielddef, m_exp, m_fieldPts...
- For a particular module, it is not clear what parts of m_f act as input and output for the module

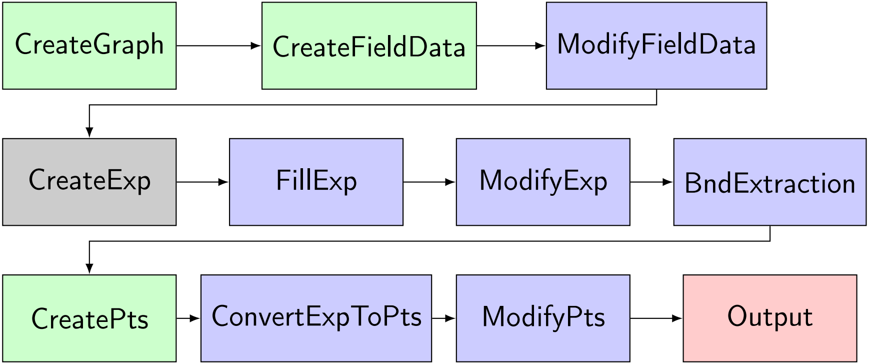


- Since when using a sequence of modules, the inputs and outputs must match, this compromises the compatibility of different modules
- For this reason, it can be difficult to chain several process modules in a single command

Modules priorities

- To solve this problem, priorities were assigned to modules
- Each priority has a well defined input and output type. For example, modules of priority ModifyExp can only operate on the m_exp portion of m_f
- There is a clear sequence for executing modules of different priorities

Modules priorities: structure



Modules priorities

Advantages of this priorities system include:

- It is clear in which order modules should be executed. Therefore, we enforce the execution in the correct sequence, automatically correcting simple usage errors

Modules priorities

Advantages of this priorities system include:

- It is clear in which order modules should be executed. Therefore, we enforce the execution in the correct sequence, automatically correcting simple usage errors
- Because the inputs and outputs of modules in each priority level are well defined, compatibility between different modules should be improved

Modules priorities

Advantages of this priorities system include:

- It is clear in which order modules should be executed. Therefore, we enforce the execution in the correct sequence, automatically correcting simple usage errors
- Because the inputs and outputs of modules in each priority level are well defined, compatibility between different modules should be improved
- It is possible to check for errors considering only the types of modules. For example, whenever we have a module of type ModifyExp we need a CreateGraph module (corresponding to an xml input)

Other improvements

The feature/FC-refactor also contains other improvements, like:

- The boundary extraction module can now output directly to .vtu and .dat
- Tidy modules to guarantee they follow the new rules
- Bug fixes

FilterFieldConvert

FilterFieldConvert allows applying FieldConvert modules to checkpoints during the simulation

- Already available since v4.4.0
- Will also benefit from feature/FC-refactor
- The AverageFields and MovingAverage filters are derived from this filter, and therefore can also use these features
- To allow accessing modules by a Filter, all modules were moved to a new library, called FieldUtils

FilterFieldConvert

Examples:

```
<FILTER TYPE="FieldConvert">
  <PARAM NAME="OutputFile">output.vtu</PARAM>
  <PARAM NAME="OutputFrequency">100</PARAM>
  <PARAM NAME="Modules"> vorticity C0Projection </PARAM>
</FILTER>

<FILTER TYPE="AverageFields">
  <PARAM NAME="OutputFile">output.fld</PARAM>
  <PARAM NAME="OutputFrequency">100</PARAM>
  <PARAM NAME="Modules"> extract:bnd=0 </PARAM>
</FILTER>
```

FilterFieldConvert

Possible reasons for using this filter:

- Convenience
- Simulations with large memory usage
- Modules leading to output files with reduced size

Summary

- A new priority system was developed in FieldConvert, improving compatibility of different modules and dependency checking
- FilterFieldConvert allows processing checkpoint files during the simulation

Thank you!