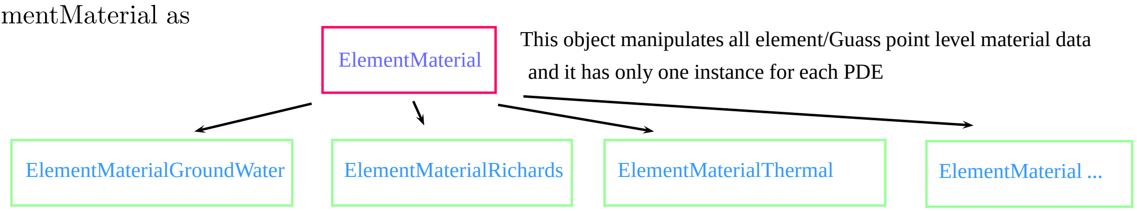
Element material object

WW

March 13, 2014

A proposal of a material object for element or integration point level material data, ElementMaterial as


ElementMaterial
This object manipulates all element/Guass point level material data
and it has only one instance for each PDE
ElementMaterialGroundWater
ElementMaterialRichards
ElementMaterialThermal
ElementMaterial ...

The class definition of ElementMaterial should include the data buffer for local level material data from the input data, and methods to calculate the coefficients of the corresponding PDE.

Purpose

To distinguish the input data and the data used for the local assembly, while to keep every needed data for material in local level of the local assembly. How to store input material data is out of this topic.

Usage

1. Create only one instance of ElementMaterial for each PDE to be solved at the beginning of the program with proper memory allocation for element or integration point level material parameters, e.g.

```
  ElementMaterialRichards _elem_mat_richards(...);
```

_elem_mat could be a member of a PDE related class or a value for the argument of a global assembly function.

2. For each element in a global assembly, fetch material data from he place they stored, whether element-wise vector or group-wise vector, to fill the variables of the ElementMaterial instance. To this purpose, the local assembly function should look like this:

```
.. template <typename T_ELE_MATERIAL, ... > localAssembly(T_ELE_MATERIAL  &elem_mat, ... )
{
      for_all(gauss_points)
      {
           ..
           ..
           // e.g. for mass matrix
           elem_mat.calMassCoef(..);
           ..
      }
}
```

and the code for global assembly should looks like, e.g.

```
 globalAssemblyRichards(_elem_mat_richards, ...  )
{
     for_all(elements)
     {
       ...
       // Get the input data stored in whichever types, element-wise or group-wise
       // and fill the element material data.
       _elem_mat_richards.setData(...);

       // Perform local assembly with a fucntion or explictily with a loop here after.
       localAssembly(_elem_mat_richards ... );
       ..
     }

}
```