Check-in

Weixiong Zheng

May 1, 2018

1 Updates on previous goals

Last week's goal was to move on restoration while prepare things for Alex.

2 Progress up to now

Other than the ordinary part of moving on.

2.1 Preparation for Alex

The task I assigned to Alex was not simply moving things from devel to restart. The reason is that through rethinking, I redesigned the data structure of `MaterialProperties` class. To get Alex started, I prepared all the necessary data-structure change in header so that Alex can do the moving work in source file with the functions in devel and data structures I put in header file.

2.2 Simplification of data input

Original design of class and functions asks for a lot of arguments. For instance, the `EquationBase` constructor is declared as:

```
template<int dim>
EquationBase<dim>::EquationBase (dealii::ParameterHandler &prm,
    std::unique_ptr<AQBase<dim>> &aq_ptr,
    std::unique_ptr<MeshGenerator<dim>> &msh_ptr,
    std::unique_ptr<MaterialProperties> &mat_ptr);
```

This potentially causes issues if developers do not use cautions. So a improvement that has been done during the past weekend was putting things together using a `struct`

```
template<int dim>
```

```
struct ComputingData {
  std::unique_ptr<AQBase<dim>> &aq_ptr;
  std::unique_ptr<MeshGenerator<dim>> &msh_ptr;
  std::unique_ptr<MaterialProperties> &mat_ptr;
};
```

s.t. once `EquationBase` is implemented, it will be much simpler and more readable

```
template<int dim>
EquationBase<dim>::EquationBase (dealii::ParameterHandler &prm,
    std::unique_ptr<ComputingData<dim>> &data_ptr);
```

The same idea will be applied to containing data for functions related to iterations. For instance, there will be a `struct` called `IterationData<dim>` with declaration of

```
template <int dim>
struct IterationData {
  std::vector<dealii::Vector<double>> &sflxes_proc;
  std::vector<std::unique_ptr<EquationBase<dim>>> &equ_ptrs;
  std::unique_ptr<IGBase<dim>> &ig_ptr;
  std::unique_ptr<MGBase<dim>> &mg_ptr;
  std::unique_ptr<EigenBase<dim>> &eig_ptr;
}
```

The goodness we are gonna see at the end is the simplification of various function definitions. For instance, the `Iterations<dim>::solve_problems` is defined as

```
template <int dim>
void Iterations<dim>::solve_problems (std::vector<Vector<double>> &sflxes_proc,
    std::vector<std_cxx11::shared_ptr<EquationBase<dim> > > &equ_ptrs,
    std_cxx11::shared_ptr<IGBase<dim> > ig_ptr,
    std_cxx11::shared_ptr<MGBase<dim> > mg_ptr,
    std_cxx11::shared_ptr<EigenBase<dim> > eig_ptr) {...}
```

After implementing the proposed struct, we are having a much simpler function argument set

```
template <int dim>
void Iterations<dim>::solve_problems (
    std::unique_ptr<IterationData<dim>> &itr_data_ptr) {...}
```

Hooray!

3 Things you need from Rachel

4 Goals/Things will be going on

I will continue whenever I have time this week as this is a moving week. Also I will see if Alex asks for help.

The other important thing is to finish up the first draft of introduction slides with equations.