Matthias Mayr 07/06/2010

Instructions, how to use SimControl.m and Input Files:

1 Introduction

This document describes, how the control routine `SimControl` works and what it is good for. In addition, it explains how you can set up your own examples.

2 SimControl

2.1 Used modules

The routine `SimControl` controls the work flow in an xfem-computation. First, a brief overview of the used modules.

2.1.1 comp_geo

`comp_geo` generates the background mesh. Input parameters for this module are saved in `xfeminputdata_comp_geo.mat`. The parameters are:

`IFmeshstructure`	mesh structure (structured, unstructured or GMSH)
`IFshapegeometryID`	shape of geometry
`IFlength`	length of rectangle
`IFheight`	heigth of rectange
`IFnldivx`	number of elements in x-direction
`IFnldivy`	number of elements in y-direction
`IFfilename_msh_file`	filename for mesh-file with mesh data from GMSH
`IFdatasetp`	dataset for matrix p (contains coords of centroid of grains)
`IFboundarydescription`	Matlab-script, that determines the nodes on the boundary domain (these may vary with different shapes of the domain)

Different matrices p are stored in `comp_geo\vdata_multi.m`. The main routine of `comp_geo` is `main_comp_geo.m`. The results (mesh data) are saved to the file `my_new_mesh.mat`. The result file has to be copied and pasted to the `preprocess` directory.
If the Neumann boundary conditions are given as a function and a structured mesh is used, the code has to determine the global node IDs of all boundary nodes by itself. This is done in the script, given by `IFboundarydescription`.

2.1.2 preprocess

The main routine `main_preprocess.m` manages the appliance of Dirichlet and Neumann Boundary Conditions (DBCs and NBCs) and assigns material properties to each element. Input data are load from `xfeminputdata_preprocess.mat`. These parameters are:

`IFDirichletBCs`	ID for a set of DBCs
`IFNeumannBCs`	ID for a set of NBCs
`IFneumann`	Way of giving the NBCs
`IFMatSet`	ID for a set of material properties

The IDs `IFDirichletBCs` and `IFNeumannBCs` refer to filenames of scripts, which describe the Dirichlet and Neumann boundary conditions for each example. The connectivity between the IDs and the files is listed in the Input File and is coded in preprocess/applybcs.m. There are two ways to describe NBCs. Using the first one, means, that the integration has to be done by the user and only nodal values are assigned. The second one enables the user to give a funtction (constant, linear, parabolic), which will be distributed on the nodes via gauss integration. The switch `IFneumann` is used to choose between these two methods.

The mesh `my_new_mesh.mat` is load, too. The boundary conditions are stored in separate files for each kind and each example, located in the folder `preprocess\Boundary_Conditions`. The material data are stored in `MaterialProperties.m`.

Results of this routine are stored to `my_new_mesh_with_BCs.mat`. The result file has to be copied and pasted to the XFEM directory.

2.1.3 XFEM

The main routine `main_xfem.m` manages the assembly and solution process. Some prostprocessing is done, too. Input data is load from `xfeminputdata_xfem.mat`. These parameters are:

`IFmethod`	choose the method to enforce constraints at interface
`IFpenalty`	penalty paramter
`IFnitsche`	stabilization parameter (for penalty terms in Nitsche's method
`IFsliding_switch`	inidcates, wheter and which kind of sliding is admitted
`IFyieldstress`	yield stress for perfect plasticity
`IFSolverType`	explicit (0) or implicit (1) solver (not needed anymore)
`IFmaxiter`	maximum number of iterations for implicit solver
`IFconvtol`	convergenz criteria for implicit solver
`IFtime`	vector with pseudotime-steps for loadstepping scheme

With IFmethod, you can choose the method to enforce the constraints at interface. There are three possibilities: (0) Lagrange multipliers, (1) penalty method, (2) Nitsche's method. The penalty paramter is set by IFpenalty. The penalty terms in Nitsche's method are not necessary to enforce the constraints. So the "penalty" paramter is only a stabilization parameter and hast to be chosen much smaller than in the penalty case. So for Nitsche's method, the penalty paramter is replaced by IFnitsche.
If IFsliding_switch is set to perfect plasticity, a yieldstress has to be specified via IFyieldstress.
Since plasticity or friction makes the problem nonlinear, a loadstepping scheme is applied. The pseude-time is given in IFtime. It is normalized to "1", that means, that the maximum load is scaled with an factor bewteen 0 and 1. The length of IFtime sets the number load steps.
The mesh with boundary conditions (my_new_mesh_with_BCs.mat) is load, too. After the computation, you can run some scripts to visualize the results: showmesh.m, showdeform.m, showdeform2.m, xx_stress.m

2.2 Operating mode of SimControl.m

SimControl.m initializes some variabeles to specify the example, which has to be solved, and then calls the three routines main_comp_geo.m, main_preprocess.m and main_xfem.m in sequence. It also manages the copying process of the result files between the subdirectories. You can specify, which of the three modules shall be executed. Perhaps, you do not want to run the background mesh generation each time, when you only changed some boundary conditions.
To choose the example, you want to solve, you have to give the filename of the appropriate input file to the variable filename_input_file without the file extension .m. To start the simulation, the current working directory has to be the one, in which SimControl.m is saved. Just run the script SimControl.m to start the computation.

3 Input Files

The data to define the examples and the code to solve the examples are quite separated. The databases are included in the code, the selection of the data is outsourced to the input file.
Input files are scripts, that initialize some variables, which configure your example. A full list and description of these variables and IDs can be found in InputFileRoutine.m.
Important: Always start a new example in InputFileRoutine.m, so that the ID list is always up-to-date in this file.
To set up a new example, open InputFileRoutine.m and configure all IDs the way, you want (Remark: Input parameter names are indicated with IF at the

beginning of the variable's name.). After setting the parameters, do not forget to update the databases, from which data shall be load. These database are

a) `preprocess\MaterialProperties.m`

   Add additionl cases to the `switch-case`-structure.

b) `preprocess\applybcs.m`

   Add additional cases to the `switch-case`-structures for DBCs and NBCs.

c) `preprocess\Boundary_Conditions`

   Make new files for DBCs and NBCs. Indicate them with an appropriate file name and the ending `_DBC.m` or `_NBC.m`.

After running a first simulation with the input file InputFileRoutine.m to check, if it works properly, saved it as `inp_NAME#_MSHX_MSHY.m`, where `NAME` is the name of the example, `#` is a number, `MSHX` is the number of elements in x-direction and `MSHY` is the number of elements in y-direction. Save this file to the same directory as the control routine `SimControl.m`.

Now, you can run this example by choosing its file name in the `SimControl` variable `filename_input_file`.