Brief introduction to parallel Gtecton

Lukas van de Wiel

November 4, 2014

Abstract

This document describes the changes from the single-processor gtecton to the May-2014 release. It is aimed at the reader familiar with Gtecton and who wishes to use the parallel version. When familiarity with Gtecton is not sufficient to understand this document, the reader is refered to the Gtecton CookBook.

1 Introduction

After five years of hard work, the parallel version of Gtecton is finally here. To use the parallelism, a few changes had to be made to the user interface. This document will list them.

2 Element/Vertex indices

First of all, the formats have been changed. The element numbers and vertex numbers have been increased to 12 digits. This is for all input and ouput files, including those defining the mesh.

3 Partitioning the mesh

Each processor of a Gtecton run needs to know what part of the mesh has been assigned to it. To accomplish this, a mesh partitioner based on METIS has been developed. Its inputs are a classic *tecin.dat.bcs* and *tecin.dat.elm*. Using the syntax:

```
partition -n tecin.dat.nps -e tecin.dat.elm -p <n_parts> -f -d <n_dimensions>
```

the files *tecin.dat.partf.bcs* and *tecin.dat.partf.elm* are created, which can be inserted in *TECIN.DAT* in the usual place. Moreover, a file *partition.f* which is used during the run.

Note that the partioning also needs to be executed for a single processor run, although it is a dummy partitioning.

4 Input files

4.1 TECIN.DAT

The format change of 12 digits for the number of vertices and the number of elements has rendered the option to enter a '-1' for the number of either to go from 5 digits to 8 digits obsolete. This syntax is no longer accepted. More over, the various 'increment' fields, for various types of boundary conditions, and for the materials, is no longer required.

4.1.1 list of vertices

The list of vertices here should be the on generated by *partition*. Its format is:

I5	I12	[2/3]G14	I5	nI12
partition	index	coordinates, [x,y(,z)]	# neighbours	neighbour indices

4.1.2 list of elements

The list of elements here should be the on generated by *partition*. Its format is:

I5	I12	4I12
partition	index	vertex indices

5 Running the simulation

A few extra command arguments have been inserted. The proper syntax for a single processor run is now of the form:

```
<f3d/pln> [as/bi] workpath=<path> partinfo=partition.info fein=TECIN.DAT fedsk=fedsk.serial
```

1. A 'workpath' had to be added. This directory points to where the data of the model is located, and where the output is to be written. Runs on a cluster may be assigned to any number of nodes, which may not be related to the main node that holds the model data. For Gtecton on those distant nodes to be able to find the model data, an explicit pointer is needed.
2. 'fein' indicates the TECIN.DAT file. This file can now have any other name as well.
3. 'partinfo' points at the partition.info file generates by *partition*
4. 'fedsk' indicates the root of the output file. Fedsk files will be generated per time step and for each processor.

Note that *partinfo*, *fein* and *fedsk* are all assumed to be in the *workpath*
For a multiple processor run, this changes to:

```
mpirun -np <number of processors> <f3d/pln> ...
```

6 Combining the data

Every run creates many fedsk files. These can be recombined with a merge utility.
The proper syntax is:

```
mergefiles partinfo=partition.info <as/bi> fedsk=fedsk.par
```

with 'fedsk' once more indicating the root of the output files. This single command handles all time steps at once. The result is one fedsk file per time step.

Output from a single processor run can also be 'merged'. Effectively, only the filename is changed in that case.

7 Plotting 3D data

Plt3d has gained the extra option 'write', which creates a *vtu* file, which can be read in Paraview, so that the output of the simulation can be plotted. A filename may be given as optional argument, such as:

```
write Flexure.vtu
```

When the file argument is ommited, the output is written to the default file *output.vtu*. For details about the *vtu* format, see `http://www.vtk.org/VTK/img/file-formats.pdf`