ParOSol User's Guide

Cyril Flaig

4. June 2012

Contents

1 Name

ParOSol

2 Synopsis

```
parosol [options] input-file
```

3 Description

ParOSol is a fully-parallel micro-FE analysis code based on octrees to solve linear elasticity problems. ParOSol solves the problem directly on 3D images, which can be obtained by CT-scans. It can be run on a desktop machine as well as on a cluster.

4 Using ParOSol

If no options are given ParOSol solves up to a tolerance of 1e–6 and uses a maximum of seven levels. An input file has to be provided. ParOSol writes the result back into the input file once the solution is complete.

A sample command is given below:

```
parosol --level 3 --tol 1e-4 simple.h5
```

This solves the elasticity equation on mesh given in the file `simple.h5`. It uses maximal three multigrid levels and solves with a relative residual of 1e–4.

5 Converting this Document

The markdown syntax is used in this user's guide. It can be translated into HTML or into other format with Pandoc. For conversion into HTML use following command:

```
pandoc -r markdown -t html -o userguide.html -c parosol.css \
       -S -N -s --toc userguide.mkd
```

Converting to LaTeX:

```
pandoc -r markdown -t latex -o userguide.tex --template=parosol.tex \
       -S -N -s --toc userguide.mkd
```

6 Getting the Source Code

The source code is available at bitbucket. To clone the repository use the following command:

```
hg clone https://bitbucket.org/cflaig/parosol
```

The source tree consists of the following directories:

- `build` directory to build ParOSol. It includes a sample configuration file
- `doc` holds the user's guide
- `src` holds all source code
- `tools` includes some non documented helper tools for generating meshes.

When extracting the file `mesh.tar.bz2` a directory will be created that includes some example meshes.

7 Building and running ParOSol

This section describes how to build ParOSol from source and to run an example mesh.

7.1 Dependencies

ParOSol has minimal dependencies on other libraries. This simplifies the installation process. To build ParOSol HDF5 library 1.8 and eigen v2 has to be installed. An MPI-compiler has to be used. OpenMPI and MPICH2 are supported.

7.2 Compiling ParOSol

If all libraries are installed from the Linux distribution, ParOSol can be built in following way:

```
cd build/
./do_configure
make
```

The MPI compiler can be set with the following variable:

```
-DCMAKE_CXX_COMPILER:FILEPATH=mpic++
```

If HDF5 is not installed in the standard library path then set the following variables:

```
HDF5_LIBRARIES:PATH=
HDF5_INCLUDE_DIRS:PATH=
```

Eigen is a pure template library. This means the library consists of header files only. This can be easily included by the header path switch (-I for *GCC*). If eigen2 is installed set following variable:

```
CMAKE_CXX_FLAGS:STRING=-I/path/toeigen
```

In the directory `build/` there is a sample configure script called `do_configure.sh`. The script can be executed with following commands:

```
cd build
./do_configure.sh
```

Instead of using the `do_configure.sh` script, **cmake** can be called directly. If the **HDF5** library is installed at `/path/to/hdf5/` the following command should be used to configure ParOSol:

```
cmake \
    -D CMAKE_CXX_COMPILER:FILEPATH=mpic++ \
    -D CMAKE_BUILD_TYPE:STRING=RelWithDebInfo \
    -D HDF5_LIBRARIES:PATH=/path/to/hdf5/lib/libhdf5.so\
    -D HDF5_INCLUDE_DIRS:PATH=/path/to/hdf5/include \
    ../src
```

After running the configure script, ParOSol can be built by simply running the following command:

```
make
```

Now the software is built in the `build/` directory.

7.3 Running a Simple Mesh

In the main directory of ParOSol there is a tarball with sample meshes. Extract it with:

```
tar xjvf mesh.tar.bz2
```

Now change the directory:

```
cd build/
```

The following command starts the simulation:

```
mpirun -np 4 ./parosol --level 3 --tol 1e-7 ../mesh/h5/sphere.h5
```

ParOSol is started with four processes. This is suited for a computer with four cores. At most three level are used in the multigrid preconditioner and the solving tolerance is set to 1e–7. For large input mesh more levels should be used. The results is written back into the original mesh.

7.4 Visualizing the Results

ParOSol ships with some helper tools. These are located in the directory `tools`. There is also a python script, `createxmf.py`, that generates a xmf file. The xmf file describes how to read the h5 mesh file. It is important that the tool is executed from the place where the meshes are stored.

```
cd mesh/h5
../../tools/createxmf.py sphere.h5
```

The newly generated file can be opened with Paraview

8 Building the Source Code Documentation

The source code of ParOSol is documented with Doxygen. The creation of the documentation is included in the build system. If Doxygen is installed and found by cmake you can run following command to build the documentation:

```
make docu
```

9 File Format

ParOSol reads and writes into the same file. Thus the file consists of an input and output part. Because HDF5 is used the file is organized in datasets and group of datasets.

9.1 Data Types Involved

The HDF5 library handles the correct endianness of the hardware. The Table below shows the data types used in the I/O operation. Since most of the computers use the little endian format, a `double` is interpreted in little endian.

`H5T_IEEE_F32LE` has to be read as IEEE floating point number in single precision and little endian.

The table gives an overview on the used formats

native	hdf5 type
`float`	`H5T_IEEE_F32LE`
`double`	`H5T_IEEE_F64LE`
`short`	`H5T_STD_U16LE`
`long`	`H5T_STD_I64LE`

9.2 Input File Format

ParOSol reads all required data from the dataset group `Image_Data`. This group must contain following datasets:

- `Image`
  - Type: `H5T_IEEE_F32LE`
  - Size: (x,y,z)
  - Description: Holds the 3D image of size x,y,z. The values are the Young's modulus in GPa.
- `Voxelsize`
  - Type: `H5T_IEEE_F64LE`
  - Size: scalar
  - Description: Size of the voxels
- `Poison_ratio`
  - Type: `H5T_IEEE_F64LE`
  - Size: scalar
  - Description: The Poison's ratio of the material `[0,0.5)`. Remark: Poison's ratio is constant over the whole image.
- `Fixed_Displacement_Coordinates`

  - Type: `H5T_STD_U16LE`
  - Size: (k, 4)
  - Description: This datasets describes the boundary conditions. The first 3 values are the coordinate of the boundary nodes. The fourth value describes the direction in which the node is fixed (`0 = x`, `1 = y`, `2 = z`).

- `Fixed_Displacement_Values`
  - Type: `H5T_IEEE_F32LE`
  - Size: k
  - Description: This dataset holds the displacements of the boundary condition. `Fixed_Displacement_Coordinates` indicates the position and the direction of the nodes and this dataset saves theirs displacements. This dataset must have the same number of rows as `Fixed_Displacement_Coordinates`.

Some load might also be applied on the nodes. Therefore following two datasets can optionally defined:

- `Loaded_Nodes_Coordinates`
  - Type: `H5T_STD_U16LE`
  - Size: (l, 4)
  - Description: This datasets is optional. It describes which nodes have loads on them. The first 3 values are the coordinates of the boundary nodes. The fourth value describes the direction in which the load acts (`0 = x`, `1 = y`, `2 = z`).

- `Loaded_Nodes_Values`
  - Type: `H5T_IEEE_F32LE`
  - Size: l
  - Description: This dataset is optional. It is strongly related to `Loaded_Nodes_Coordinates`. `Loaded_Nodes_Coordinates` stores the position and the direction of the loaded nodes and this dataset stores the amount of the load per node. This dataset must have the same number of rows as `Loaded_Nodes_Coordinates`.

9.3 Output file format

ParOSol writes the solution and the mesh. The mesh is needed to visualize the solution. The solution and the mesh are two different groups of datasets. The `Mesh` group contains following datasets:

- `Coordinates`

  - Type: `H5T_IEEE_F32LE`
  - Size: (n,3)
  - Description: This dataset holds the coordinates of the vertices.
- `Elements`
  - Type: `H5T_STD_I64LE`
  - Size: (m,8)
  - Description: This dataset stores the 8 indices of the adjacency nodes of an element.
- `Emoduli`
  - Type: `H5T_IEEE_F32LE`
  - Size: (m,1)
  - Description: This dataset stores for each element the Young's modulus.

The results and some post processing values are stored in the group `Solution`:

- `disp`
  - Type: `H5T_IEEE_F64LE`
  - Size: (n,3)
  - Description: Displacement at each node.
- `force`
  - Type: `H5T_IEEE_F64LE`
  - Size: (n,3)
  - Description: Force at each node.
- `SED`
  - Type: `H5T_STD_F64LE`
  - Size: (m,1)
  - Description: Strain Energy Density. It is computed at the center of the elements.
- `VonMises`
  - Type: `H5T_STD_F64LE`
  - Size: (m,1)
  - Description: Von Mises Stress. It is computed at the center of the elements.
- `EFF`

- Type: `H5T_STD_F64LE`
- Size: (m,1)
- Description: Effective strain. It is computed at the center of the elements.