GMVDIFF - A Tool For Generating Differences and Projections Of Node Data Included In GMV Files

Jens F. Acker

October 15, 2001

Fachbereich Mathematik, Angewandte Mathematik und Numerik (LS III),
Universität Dortmund
Vogelpothsweg 87, 44227 Dortmund, Germany
Email: featflow@featflow.de
Homepage: http://www.featflow.de

Contents

List of Figures

1 Motivation

The collection of our fluid simulation programs called FEATFLOW produces huge data files in GMV format. To get a better understanding of the effects caused by altering certain simulation parameters and different methods used, we needed a small and flexible tool with following properties:

- It is able to calculate the differences between different data files
- It should use only the node data provided in data files

- The used meshes should not be bound to be equal
- It should be able to restrict the differences to the nodes of a coarser mesh
- It doesn't make much assumptions on the node data fields provided
- It should be fast enough to process huge meshes

2 Command line syntax of GMVDIFF

If you call GMVDIFF with no or false parameters a short help message with the command line syntax will be displayed. Essentially GMVDIFF needs two gmv data files where it can grab the data from to compute the difference (the second one can be substituted with zero data) and a gmv file containing the mesh on which to project the data. Compressed data and mesh files can also be used (they will be decompressed to /tmp/ and later deleted, allowed are .gz, .Z and .bz2 files). In the resulting file, there will be a reference to the mesh file for node and cell data to save space.

```
Usage: gmvdiff [-f] [-2] data1 [-z|data2] mesh
```

-f Fast mapping. The node numbers in the specified mesh are still valid in the two data files. This option gives a gain in speed and memory usage and should be used whenever possible.

-2 Assume a 2D mesh aligned at the XY-plane. This option gives a minor speedup in cases when the **-f** option cannot be used.

-z Assume a data file with zero data in all expected data fields as second file. You have to omit the second data file on the command line.

3 How Does It Work Internally ?

Essentially GMVDIFF reads first the node coordinates of the meshes and tries to find the node numbers of each node of the target mesh in the source meshes. Then it computes the differences for each node found and each data field existent in both data files and writes the data to the standard output in GMV format. In the default case GMVDIFF builds an oct-tree or quad-tree (**-2** option) for each mesh contained in the data files and tries to find the nodes of the target mesh by looking them up in the oct-tree. With that algorithm there is only an $m \cdot \log_{10} n$ complexity instead of an $m \cdot n$ complexity. Even for medium sized meshes with $m = 1000$ nodes to find in $n = 10000$ nodes you have a 4000 versus 10^7 advantage. This step is unnecessary if the node numbers of the target mesh nodes are still the same in the meshes contained in the data files and can be switched off by the **-f** option.

A catch and at the same time a nice side effect of the oct-tree algorithm is the fact that geometrical equivalent nodes cannot be separated and the algorithm doesn't terminate. This kind of loop will be caught by GMVDIFF and the equivalent nodes will be displayed on the standard output. If you generate very fine meshes and do not write them out with the necessary precision, some mesh nodes can fuse geometrically. Meshes with that kind of defect are hard to detect by hand and will cause problems with other programs like GMVPT.

4 Examples

4.1 Projecting Data Onto A Coarser Mesh

This example explains how to project data from a multigrid computation to a mesh of a coarser refinement level. The source data and mesh are inside a file named 'ex1_in.gmv', shown in fig. 1, and the target mesh is inside a file named 'tria3.gmv', shown in fig. 2. To project the data and to store the result in a file named 'ex1_out.gmv' you can execute GMVDIFF with the following command line:

```
gmvdiff -fz ex1_in tria3.gmv >ex1_out.gmv
```

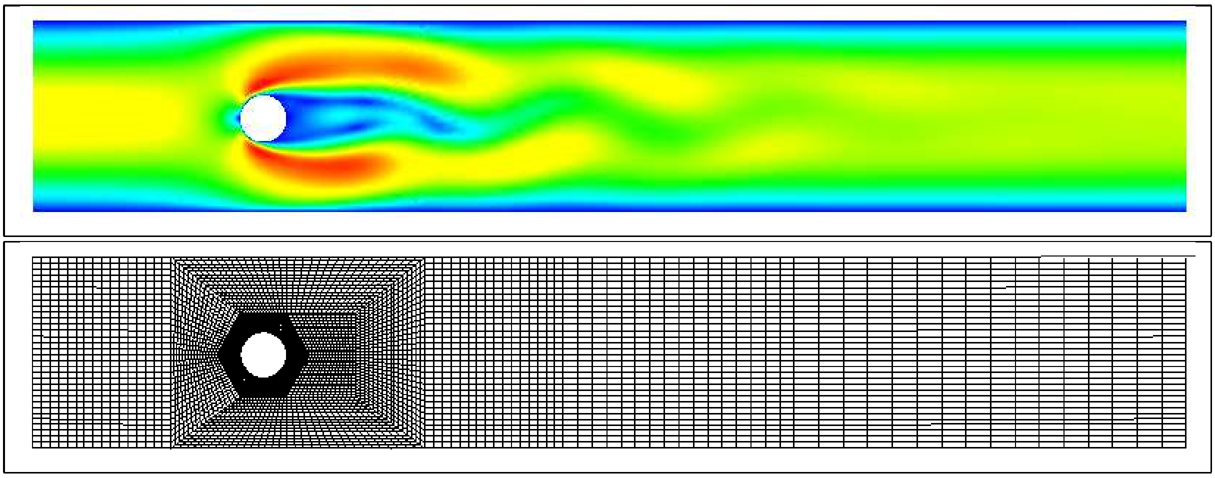

Figure 1: Original data (norm of velocity shown here) and the used mesh

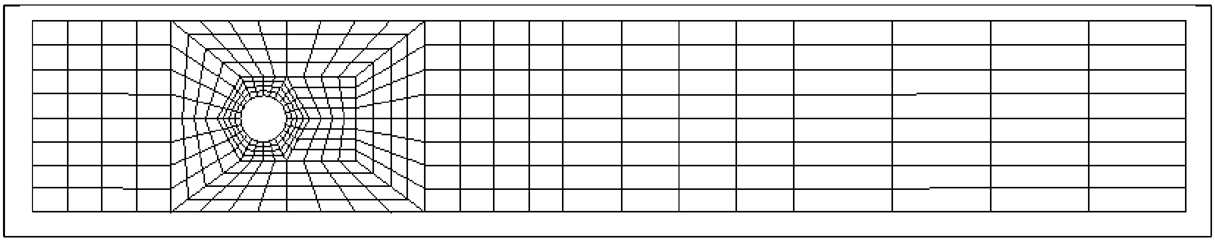

Figure 2: Target mesh

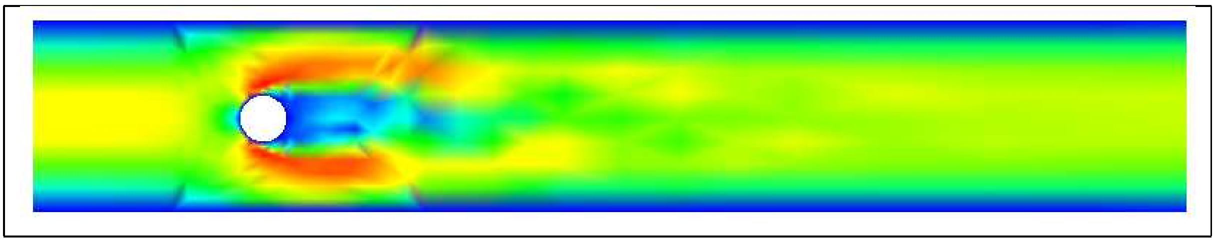

Figure 3: Output data (norm of velocity)

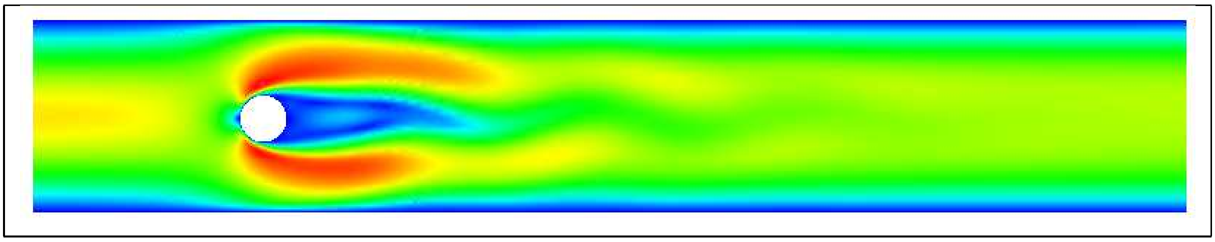

Figure 4: Original data of a new simulation (velocity norm)

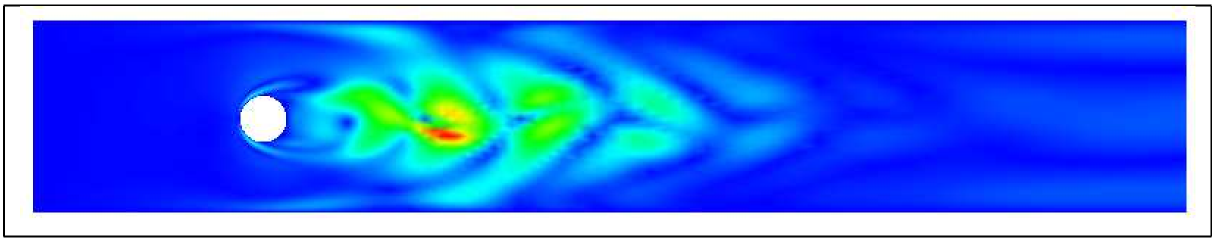

Figure 5: Data changes between the two simulations

The output data produced is shown in fig. 3.

Beware! *The string you use to specify the name of the target mesh will be put as reference into the header of the resulting gmv file without change. Therefore you can omit the extension '.gmv' for the data file but not for the mesh file.*

4.2 Comparing Data On The Same Mesh

This example explains how to use GMVDIFF to see the changes between different data files. Now we repeat the previous simulation with a different fluid viscosity and compare the new output for the same time (see fig. 4) with the old output (see fig. 1). The results can be seen in fig. 5 (the differences have a data range not related to the original data ranges). If you have your first data set in the file 'data1.gmv', the second in the file 'data2.gmv' and your mesh in 'tria5.gmv', you can execute GMVDIFF with the following command line:

```
gmvdiff -f data1 data2 tria5.gmv >diff2.gmv
```

4.3 Comparing Data On Overlapping Meshes

Next we will explain how to compare data on meshes that share only a subset of nodes and/or have a completely different node numbering. First we add an additional column of cells to the simulation domain of our previous examples and repeat the simulation (see fig. 6). GMVDIFF computes the difference in the nodes shared by the two input data files and the target mesh. Nodes in the target mesh which are not contained in one input data file will be assigned the value zero for all data fields in that specific input file before the difference will be computed. This can lead to some odd visual effects in the output file. We cannot use the '-f' switch here, because the numbering cannot be reused between the used meshes. If the mesh file ist named 'mesh.gmv' and the data files are named 'data1.gmv' and 'data2.gmv', you can execute GMVDIFF with the following command line:

```
gmvdiff data1 data2 mesh.gmv >diff3.gmv
```

The produced output (a file named 'diff3.gmv') is show in fig. 7.

5 Summary

It was shown, that GMVDIFF can be used for various purposes like projection between different refinement levels or overlapping meshes, and comparisions between different simulations on the same mesh. Even some nasty mesh

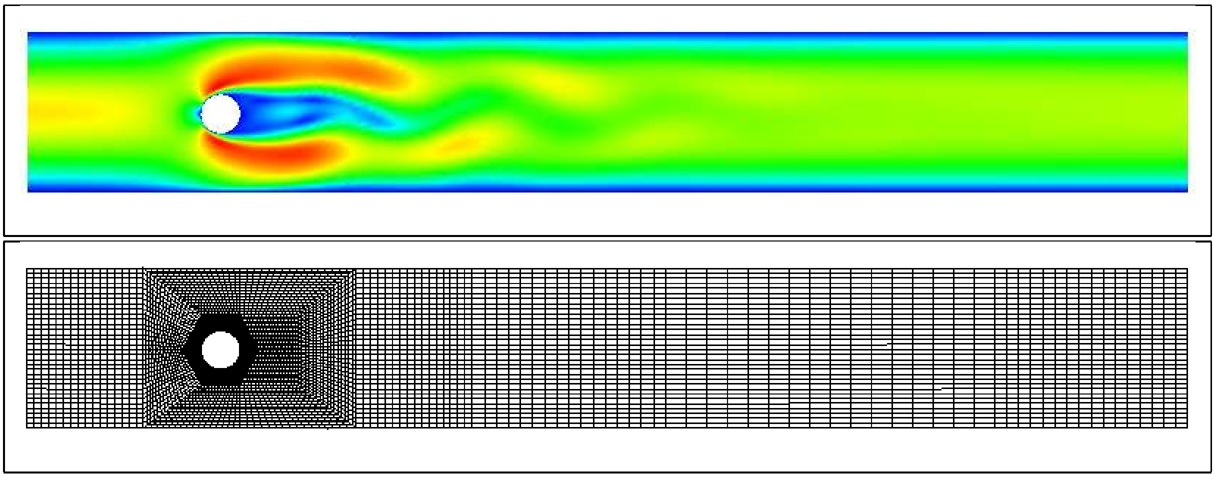

Figure 6: Expanded simulation domain (velocity data) and the used mesh

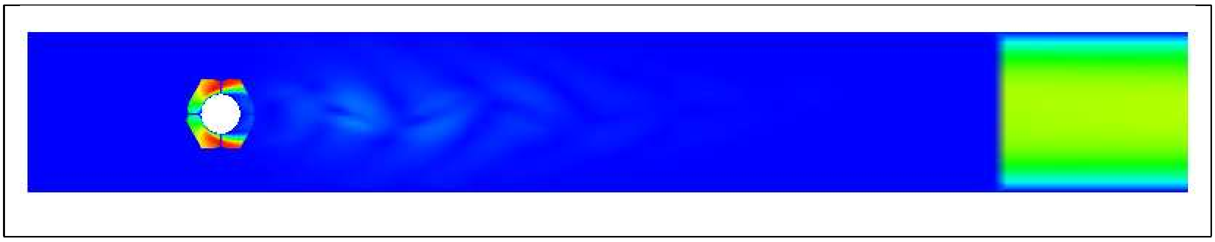

Figure 7: Data differences on shared nodes

errors will be detected. Additionally all these tasks will be performed with maximum efficiency, consequently fulfilling all previous desired properties.