



USER MANUAL
BIOMEDICAL APPLICATION

WP1 – Task 1.1

Document Filename:	CG1.1-v1.0-UvA007-BiomedicalAppUserManual.doc
Work package:	WP1 – Task 1.1
Partner(s):	UvA, University Linz
Lead Partner:	UvA
Config ID:	CG1.1-v1.0-UvA007-BiomedicalAppUserManual
Document classification:	PUBLIC

Document Log

Version	Date	Summary of changes	Author
0.1	20/11/2004	Draft version	Alfredo Tirado-Ramos
0.2	25/11/2004	Added GVK-BFVS description	Herbert Rosmanith, Paul Heinzlreiter
1.0	25/11/2004	Final revision	Alfredo Tirado-Ramos
	26/01/2005	Verified by the QE	Robert Pajak

CONTENTS

1. COPYRIGHT NOTICE	4
2. INTRODUCTION.....	5
2.1. ABBREVIATIONS AND ACRONYMS	5
2.2. REFERENCES	5
3. BLOODFLOW SIMULATOR USAGE.....	6
3.1. RUNNING THE PRODUCT	6
3.1.1. <i>Step-by-Step User Setup</i>	6
4. BLOODFLOW VISUALIZATION SERVICE USAGE	7
4.1. BFVS ARCHITECTURAL OVERVIEW	7
4.2. INSTALLATION AND CONFIGURATION	7
4.3. STARTING GVK-BFVS.....	7
4.4. PROCESSING THE OUTPUT.....	9
4.5. PLUGIN PARAMETERS	9
5. CONTACT INFORMATION AND CREDITS.....	11
6. THE EDG LICENSE AGREEMENT	12

1. COPYRIGHT NOTICE

Copyright (c) 2005. All rights reserved.

Use of this product is subject to the terms and licenses stated in the EDG license agreement. Please refer to Section 4 for details.

This research is partly funded by the European Commission IST-2001-32243 Project “CrossGrid”.

2. INTRODUCTION

The B_Stream1.1 is a full-blown parallel, 3D implementation of the D3Q19 Lattice Boltzmann method. B_Stream1.1 allows simulation of a time-periodic flow in a tube, stenosis or bifurcation for different Reynolds (Re) numbers.

This code is for use within the CrossGrid project:

1. to be used as an application kernel in Grid Bench,
2. to be used as a test case for tools such as Marmot and G-PM.

The GVK-BFVS blood flow visualization service features an offline rendering approach for pre-computed polydata. The pre-computation takes place on the grid, while the rendering takes place on the local workstation.

2.1. ABBREVIATIONS AND ACRONYMS

B_Stream1.1	Blood flow stream solver
Re	Reynolds number
G-PM	Grid Performance Monitoring
GVK	Grid Visualization Kernel
GVK-BFVS	Grid Visualization Kernel for Bloodflow Visualization
VTK	Visualization Toolkit
QT	C++ Framework for building GUIs
DesktopVRE	Desktop Virtual Radiology Explorer
UvA	University of Amsterdam
JKU	Johannes Kepler University
CVS	Concurrent Versions System
TAR	Tape Archiver

2.2. REFERENCES

QT is available from <http://www.trolltech.com>

VTK is available from <http://public.kitware.com/vtk>.

GVK-BFVS plugin binary is available from <http://clio.gup.uni-linz.ac.at/md/plugin/GVKBFVS.jar>.

3. BLOODFLOW SIMULATOR USAGE

3.1. RUNNING THE PRODUCT

B_Stream1.1 needs the value of Re-number and the input files "filename.conf" and "filename.bs". It can be run also only with "filename.conf" input file (it will produce a simple tube).

The output of the program is the set of "N_velocity.vtk" files (N is the number of iterations). These files contain the velocity vector flow, and the "Output" file, which contains all the simulation parameters and error notifications.

3.1.1. Step-by-Step User Setup

```
$ make
```

```
$ mpirun -np N B_Stream1.1 "Re" "filename"
```

To delete the output files:

```
$ make clean
```

To delete the output from compilation:

```
$ make cleanall
```

The source code for B_Stream1.1 is in the Crossgrid CVS, at:

http://cvs.fzk.de/cgi-bin/viewcvs.cgi/crossgrid/crossgrid/wp1/wp1_1-biomed/B_Stream1.1/

4. BLOODFLOW VISUALIZATION SERVICE USAGE

4.1. BFVS ARCHITECTURAL OVERVIEW

The GVK-BFVS requires two parts: The simulator, built by the UvA, and the visualisation routines, and methods for visualising the simulator output, built by UvA and JKU as a common effort. The source is available via CVS from the Crossgrid repository. Local rendering is integrated into the DesktopVRE, the Virtual Radiology Explorer, available from the UvA.

4.2. INSTALLATION AND CONFIGURATION

GVK-BFS requires:

- Java \geq 1.4.2
- QT \geq 3.3.2
- VTK \geq 4.5.0

Java is needed by the plugin for the Migrating Desktop. QT is needed by DesktopVRE (local renderer). VTK is needed by both the DesktopVRE and the GVK.

It's necessary to copy the required files to the "Virtual Directory" of the Migrating Desktop. From there, the Migrating Desktop will copy the files to whichever execution location is requested.

Files:

- gvkbfvs.sh - a bourne shell script, required for parameter passing between gvk programs
- gvk-compute-pathlines - a C program, accepting as input the CFD data and computing pathlines. It requires VTK.
- gvk-pathlines2polydata - a C program for converting pathlines to polydata. It requires VTK.

The output that GVKBFVS produces is stored in the Migrating Desktop File named "StdOutput", which could e.g. be located in the Virtual Directory. From there, the user can transfer it to the local workstation and use the DesktopVRE to view it.

4.3. STARTING GVK-BFVS

GVK-BFVS is start from within the Migrating Desktop. From the "Tools -> Job Wizard" Tab select "applications". From there, select "GVK blood-flow visualisation". The following plugin window will pop up:

Job Submission Wizard

Arguments Description Resources **Files** Environment Tools

GVK - BFVS

Input File Prefix:

Output File Prefix:

Test File:

Number of Loops:

Maximum Propagation:

Initial Integration Step:

Number of Frames:

Number of Particles:

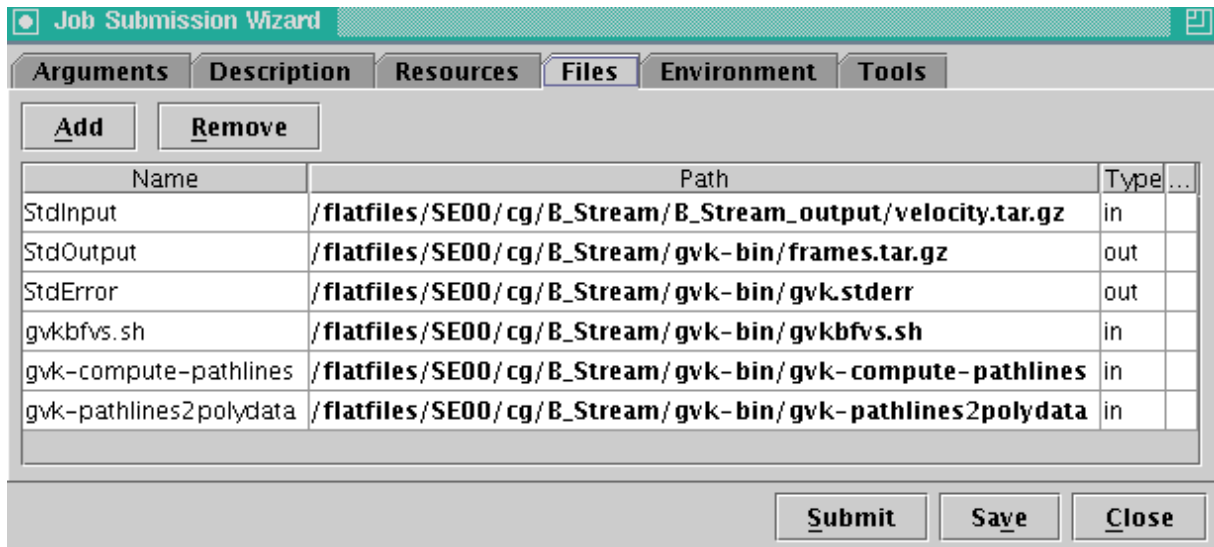
ShutterSpeed:

Submit Save Close

In the "Files" tab of the plugin, the following files need to be defined:

- StdInput - location of the data that the CFD produced
- StdOutput - location of polydata, ready for local rendering
- StdError - misc. error information
- gvkbfsh.sh, gvk-compute-pathlines, gvk-pathlines2polydata: programs required for GVK-BFVS operation (see above).

The user should copy the required programs to the Virtual Directory using the "Grid Commander" (integral part of the Migrating Desktop). An example can be seen in the following screenshot:



The “/flatfiles” path is some location on a storage element.

4.4. PROCESSING THE OUTPUT

After GVK-BFVS has finished its operation, the Grid Commander can be used to transfer resulting tar file to the local workstation. It should be noted that for some reasons, GVK-BFVS produces a single, compressed TAR file. Locally, this file will usually be unpacked before viewing.

4.5. PLUGIN PARAMETERS

A general note about file parameters: the CFD/GVK operate on a set of files. Transferring a set of files is not supported by the Migrating Desktop. Therefore, file sets are merged and transferred as TAR files.

- Input File Prefix - directory/prefix settings. This parameter contains the path & file prefix where GVK-BFVS will temporary store its files for computation.
- Output File Prefix - directory/prefix settings. this parameter contains the path & file-prefix where GVK-BFVS will temporary store its files after the pathlines computation.
- Test File - a single frame (file) of output from the CFD. this is a reference file required to produce pathlines.
- Number of Loops - how often do we iterate over the pathline computation.
- Maximum Propagation
- Initial Integration Step
- Number of Frames - determines how many pictures to produce.
- Number of Particles - determines how many blood particles to add in each pathline computation.

- Shutterspeed - compare this to the shutter of a camera: the longer time moving car will be seen as line; the same applies for the visible path which blood particles put back in a particular frame.

The source code of the plugin is in the Crossgrid CVS. Its name is GVKBFVS.java, binary: GVKBFVS.jar. The plugin binary is downloaded from its homepage at:

<http://clio.gup.uni-linz.ac.at/md/plugin/GVKBFVS.jar>.

5. CONTACT INFORMATION AND CREDITS

Coordination & management:

P.M.A. (Peter) Sloot (Scientific Lead) <sloot@science.uva.nl>

A.G. (Alfons) Hoekstra (WP1 Lead) alfons@science.uva.nl

E.V. (Elena) Zudilova (WP1.1 Lead) elenaz@science.uva.nl

D. (Dieter) Kranzlmüller (GVK Lead) kranzlmuller@gup.jku.at

Research and development team:

A. (Alfredo) Tirado Ramos (System architect) <alfredo@science.uva.nl>

D.P. (Denis) Shamonin (DesktopVRE) <dshamoni@science.uva.nl>

R.M. (Roman) Shulakov (PSS) <rshulako@science.uva.nl>

D.J. (Derek) Groen (UvA System Administrator) <djgroen@science.uva.nl>

Z. (Zhiming) Zhao (HLA) <zhiming@science.uva.nl>

P. (Paul) Heinzlreiter (GVK-BFVS Coordination) <heinzlreiter@gup.jku.at>

H. (Herbert) Rosmanith (GVK-BFVS Developer) <rosmanith@gup.jku.at>

6. THE EDG LICENSE AGREEMENT

Copyright (c) 2005 CrossGrid. All rights reserved.

This software includes voluntary contributions made to the CrossGrid Project. For more information on CrossGrid, please see <http://www.eu-crossgrid.org>.

Installation, use, reproduction, display, modification and redistribution of this software, with or without modification, in source and binary forms, are permitted. Any exercise of rights under this license by you or your sub-licensees is subject to the following conditions:

1. Redistributions of this software, with or without modification, must reproduce the above copyright notice and the above license statement as well as this list of conditions, in the software, the user documentation and any other materials provided with the software.

2. The user documentation, if any, included with a redistribution, must include the following notice: “This product includes software developed by the CrossGrid Project (<http://www.eu-crossgrid.org>).”

Alternatively, if that is where third-party acknowledgments normally appear, this acknowledgment must be reproduced in the software itself.

3. The names “CrossGrid” and “CG” may not be used to endorse or promote software, or products derived therefrom, except with prior written permission by cgooffice@cyfronet.krakow.pl.

4. You are under no obligation to provide anyone with any bug fixes, patches, upgrades or other modifications, enhancements or derivatives of the features, functionality or performance of this software that you may develop. However, if you publish or distribute your modifications, enhancements or derivative works without contemporaneously requiring users to enter into a separate written license agreement, then you are deemed to have granted participants in the CrossGrid Project a worldwide, non-exclusive, royalty-free, perpetual license to install, use, reproduce, display, modify, redistribute and sub-license your modifications, enhancements or derivative works, whether in binary or source code form, under the license conditions stated in this list of conditions.

5. DISCLAIMER

THIS SOFTWARE IS PROVIDED BY THE CROSSGRID PROJECT AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, OF SATISFACTORY QUALITY, AND FITNESS FOR A PARTICULAR PURPOSE OR USE ARE DISCLAIMED. THE CROSSGRID PROJECT AND CONTRIBUTORS MAKE NO REPRESENTATION THAT THE SOFTWARE, MODIFICATIONS, ENHANCEMENTS OR DERIVATIVE WORKS THEREOF, WILL NOT INFRINGE ANY PATENT, COPYRIGHT, TRADE SECRET OR OTHER PROPRIETARY RIGHT.

6. LIMITATION OF LIABILITY

THE CROSSGRID PROJECT AND CONTRIBUTORS SHALL HAVE NO LIABILITY TO LICENSEE OR OTHER PERSONS FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL, EXEMPLARY, OR PUNITIVE DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, LOSS OF USE, DATA OR PROFITS, OR BUSINESS INTERRUPTION, HOWEVER CAUSED AND ON ANY THEORY OF CONTRACT, WARRANTY, TORT (INCLUDING NEGLIGENCE), PRODUCT LIABILITY OR OTHERWISE, ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.