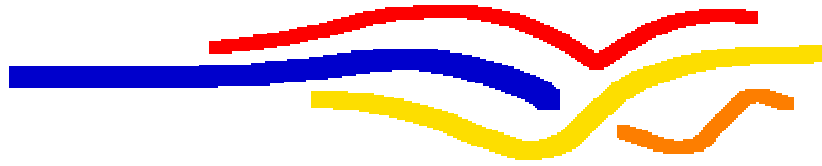


Center for Simulation of Advanced Rockets



University of Illinois at Urbana-Champaign

Rocdiff User's Guide

Center for Simulation of Advanced Rockets
University of Illinois at Urbana-Champaign
2270 Digital Computer Laboratory
Urbana, IL

Title:	Rocdiff User's Guide
Author:	Courtlan McLay
Subject:	This document describes the use of the Rocdiff utility
Revision:	Revision 1
Revision History	Revision 0: Initial Release Revision 1: Update to cover Roccom integration
Effective Date:	12/09/05

1.0 Introduction

1.1 *Rocdiff Overview*

Rocdiff is a utility for quantitatively measuring the difference between two Rocstar simulation output files. It uses the Rocin API to read a pair of files given as commandline arguments, then computes selected numerical metrics on each attribute, outputting the results in tabular format. Rocdiff has been tested with HDF4 output files from several modules, including Rocflo and Rocfrac, although in principle it should work with any pair of Rocstar output files containing volume or surface data. Since Rocdiff uses Rocin for all file-based input, it will seamlessly support CGNS format when it becomes available in Rocstar 3. Rocdiff is written in C++ and uses the Roccom framework. It will compile and link on platforms supported by Rocstar, which presently include many variants of Unix and Linux, as well as OS X.

1.2 *Related Documents*

The information in this guide is supplemented by the following documents:

- “Rocstar 3 User's Guide”.
- “Roctest User's Guide”.
- “Rocin User's Guide”.
- “Rocketeer User's Guide”.

2.0 Purpose and Methods

Rocdiff's purpose is to provide a means for comparing the solution field between two Rocstar simulations, specifically for implementing regression testing, although it may also be used as a general comparison utility for Rocstar output files. Rocdiff takes two filenames as arguments on the commandline, and begins by checking their existence and loading them into memory using the Rocin API. These files must have the same number of elements and identical connectivity information, because Rocdiff will perform a point-to-point comparison of attribute values. Rocdiff calculates up to two metrics for each attribute vector in the file, and outputs the metric results in tabular format. Ultimately, it will optionally take two directories as commandline arguments, and calculate metric tables for each pair of corresponding files in those two directories. The directory comparison functionality is not yet implemented.

2.1 *Maximum Difference Metric*

Rocstar output files may contain an arbitrary number of attributes, each of which is an array of integer or floating-point numbers. Although the Roccom API supports many other data types such as byte and character data, numerical comparisons of such data are not currently supported

by Rocdiff. Attributes can be scalar quantities, such as pressure or stress, or they can be vector quantities such as nodal coordinates or velocity. One of the simplest measures of error between attribute fields of numerical data is the maximum difference metric. This metric is computed by finding the largest absolute difference between corresponding data points in the two arrays:

$$\text{maxdiff} = \max_{I,J} |X_{ij} - Y_{ij}|$$

2.2 Mean Square Error Metric

A second metric implemented in Rocdiff for the comparison of two attribute fields is mean square error (MSE). MSE is a commonly used metric for comparing images and other scientific data. Unlike maximum difference, MSE is not as sensitive to large differences at very few points in the two arrays because it is an average (or mean) of the square of difference over all data points. This can be written:

$$MSE = \frac{1}{IJ} \sum (X_{ij} - Y_{ij})^2$$

3.0 Building and Running

Rocdiff is a small application written in C++. It uses the same makefile infrastructure that Rocstar uses, though it is currently not integrated with the Rocstar distribution under the CSAR CVS repository in the genx/Codes directory. Rocdiff sourcecode and makefiles are located in /Rocstar/Rocdiff/Codes in CVS. The structure of the sourcecode is given below:

```
Rocdiff/Codes/Makefile
Rocdiff/Codes/Makefile.basic
Rocdiff/Codes/Makefile.in
Rocdiff/Codes/src/BaseMetric.[C,h]
Rocdiff/Codes/src/main.C
Rocdiff/Codes/src/RocdiffWindow.[C,h]
```

3.1 Library Dependencies

Before building Rocdiff, you must also install and build the libraries it uses. Rocdiff links with Roccom and Rocin for managing I/O, and Rocin in turn uses NCSA's HDF4 library and/or CGNS for managing scientific file formats. See the Rocstar 3 User's Guide for instructions on how to obtain and install HDF4 for Rocstar.

Because Rocdiff is not integrated with the Rocstar distribution, the dependencies in the Roccom sourcecode hierarchy are provided via a symbolic link in the CVS repository. It is suggested that Rocdiff source tree be checked out as a whole:

```
>$ cvs co Rocstar/Rocdiff
```

This puts the sourcecode and makefiles for Roccom and Rocdiff at the correct depth relative to each other, since the Rocstar 3 makefile architecture requires that Roccom/Makefile.common location be specified in the dependent module's makefile.

3.2 Building Rocdiff

Build the application and the supporting Roccom libraries by changing your working directory to the Rocdiff/Codes directory and issuing the appropriate “make” command. See the Rocstar 3 User's Guide for details on all the Rocstar makefile options (LIBSUF, DEBUG, CGNS, etc), as well as the platforms currently supported. The default dynamically linked application, with .so libraries for both Roccom and Rocin can be compiled with the command:

```
>$ cd Rocstar/Rocdiff/Codes  
Rocstar/Rocdiff/Codes>$ gmake
```

This will produce a binary executable named “rocdiff” in the Codes directory.

3.3 Running Rocdiff

Rocdiff is invoked with a minimum of two commandline arguments, specifying the full paths to the files being compared:

```
>$ rocdiff -mdiff rocket_X/Rocflo/Rocout/fluid_09.800000_0001.hdf  
rocket_Y/Rocflo/Rocout/fluid_09.800000_0001.hdf
```

With the “-mdiff” commandline switch, the maximum difference metric will be computed by default and written to a table on standard output. The example given is for two Rocflo volume data files, and the resulting table of attributes and metric values is shown below:

FILE1	FILE2	ATTRIBUTE	MDIFF
fluid_09.800000_0001.hdf	fluid_09.800000_0001.hdf	nc[0]	1.2252e-03
fluid_09.800000_0001.hdf	fluid_09.800000_0001.hdf	nc[1]	4.9123e-07
fluid_09.800000_0001.hdf	fluid_09.800000_0001.hdf	nc[2]	4.9131e-07
fluid_09.800000_0001.hdf	fluid_09.800000_0001.hdf	siVel	3.2765e-05
fluid_09.800000_0001.hdf	fluid_09.800000_0001.hdf	sjVel	6.5334e-06
fluid_09.800000_0001.hdf	fluid_09.800000_0001.hdf	skVel	6.5234e-06
fluid_09.800000_0001.hdf	fluid_09.800000_0001.hdf	dtf	1.4646e-07
fluid_09.800000_0001.hdf	fluid_09.800000_0001.hdf	rhof	9.0397e-01
fluid_09.800000_0001.hdf	fluid_09.800000_0001.hdf	rhovf[0]	1.0936e+01
fluid_09.800000_0001.hdf	fluid_09.800000_0001.hdf	rhovf[1]	2.2489e-02
fluid_09.800000_0001.hdf	fluid_09.800000_0001.hdf	rhovf[2]	2.2476e-02
fluid_09.800000_0001.hdf	fluid_09.800000_0001.hdf	rhoEf	3.9238e+06
fluid_09.800000_0001.hdf	fluid_09.800000_0001.hdf	vf[0]	3.3203e+00
fluid_09.800000_0001.hdf	fluid_09.800000_0001.hdf	vf[1]	4.7558e-03
fluid_09.800000_0001.hdf	fluid_09.800000_0001.hdf	vf[2]	4.7534e-03
fluid_09.800000_0001.hdf	fluid_09.800000_0001.hdf	Tf	7.9329e+01
fluid_09.800000_0001.hdf	fluid_09.800000_0001.hdf	pf	8.4127e+05
fluid_09.800000_0001.hdf	fluid_09.800000_0001.hdf	af	1.3315e+01
fluid_09.800000_0001.hdf	fluid_09.800000_0001.hdf	rand	0.0000e+00

4.0 Input and Output (User Interface)

Rocdiff presently has four commandline switches for controlling its execution:

<i>Switch</i>	<i>Description</i>
-all	Compute all available metrics
-file <filename>	Specify a file to write the tabular data to (default is standard output)
-mse	Compute the mean square error metric
-mdiff	Compute the maximum difference metric

4.1 Input File Structure

Rocdiff takes two Rocin/Rocout formatted files as commandline arguments. At present, the Rocin and Rocout service modules support both NCSA's HDF4 file format, and a newer CGNS standard which is implemented on top of NCSA's HDF5 file format.

The original HDF input file format is documented in Appendix A of the Rocin User's Guide, and in more detail in the Rocketeer User's Guide. A draft specification of the CGNS file format for Rocstar is given in Appendix B of the Rocin User's Guide.

4.2 Output File Structure

Rocdiff produces tabular data for each attribute in the pair of Rocstar files being compared. If the `-file <filename>` switch is specified, a new ASCII text file named `<filename>` is created and the table is written to the file. Depending on the metric(s) selected on the commandline, the general format of the table will be:

<i>Filename1</i>	<i>Filename2</i>	<i>Attribute Name</i>	<i>Metric1</i>	<i>Metric2</i>
------------------	------------------	-----------------------	----------------	----------------

The filenames are shortened to eliminate any preceding directories. The available metrics, maximum difference and mean square error are selected by using `-mdiff` and `-mse` respectively. If no metric switch is given, the default will be just the maximum difference. Each attribute is usually an abbreviated moniker for a physical quantity. For example, in the Rocflo and Rocflu fluid dynamics volumetric data some commonly used values and their meanings are given by the following table:

<i>Rocflo/Rocflu Attribute Name</i>	<i>Description</i>
Nc	Nodal coordinate component
Af	Speed of sound
Rhof	Fluid density
Dtf	Fluid timestep
Vf	Fluid velocity component
Tf	Fluid temperature
Pf	Fluid pressure

In the Rocfrac solid mechanics module's volumetric data, some common attributes and their meanings are given below:

<i>Rocfrac Attribute Name</i>	<i>Description</i>
Nc	Nodal coordinate component

<i>Rocfrac Attribute Name</i>	<i>Description</i>
Disp	Total nodal displacement
disp_burn	Nodal displacement due to surface burning
Stress	Stress in the solid

The basic structure of an HDF file for Rocstar 3 is given in the Appendix of the Rocketeer User's Guide. A more detailed explanation of all the attribute quantities can be found in the corresponding physics module's User's Guide (Rocflo, Rocflu, Rocfrac, Rocsolid).

In the future, additional metrics will be added to Rocdiff, each with it's own commandline switch.

5.0 Examples and Test Problems

TBD