Automatic Simulocean Report

Release

user name

27. May 2013

Contents

SIMULOCEAN

CHAPTER 1

Introduction

Sample Description here.

CHAPTER 2

Background Description

Under what condition to make such a simulation. fetch formatted description model.

CHAPTER 3

Model

Fetch model description from models.

Delft3D, developed by Deltares (formerly Delft Hydraulics), is a flexible integrated modelling suite, which simulates two-dimensional (in either the horizontal or a vertical plane) and three-dimensional flow, sediment transport and morphology, waves, water quality and ecology and is capable of handling the interactions between these processes. After Delft3D-FLOW was open-sourced in 2011, more and more researchers started using Delft3D.

CHAPTER 4

Job

Fetch job input data from models and render it here.

CHAPTER 5

Output

date, time : 2013-05-24, 11:34:31 SUMMARY FOR PARTITION : 1 * **WARNING Thin dam (50, 141) lies on an inactive point 0 errors and 1 warnings returning to main program from domain new02b ———————————————————— SUMMARY FOR PARTITION : 2 0 errors and 0 warnings returning to main program from domain new02b ———————————————————— SUMMARY FOR PARTITION : 3** * WARNING Dry point (7, 113) lies on an inactive point *** WARNING Station lies outside the computational domain 0 errors and 2 warnings returning to main program from domain new02b ————————————————————

This shows you the available output data

FINISHED Delft3D-FLOW runid : new02b date, time : 2013-05-24, 11:34:31 SUMMARY FOR PARTITION : 1 * **WARNING Thin dam (50, 141) lies on an inactive point 0 errors and 1 warnings returning to main program from domain new02b ———————————————————— SUMMARY FOR PARTITION : 2 0 errors and 0 warnings returning to main program from domain new02b ———————————————————— SUMMARY FOR PARTITION : 3** * WARNING Dry point (7, 113) lies on an inactive point * **WARNING Station lies outside the computational domain 0 errors and 2 warnings returning to main program from domain new02b ———————————————————— SUMMARY FOR PARTITION : 4** * WARNING Station lies outside the computational domain 0 errors and 1 warnings returning to main program from domain new02b ———————————————————— D_Hydro [1369413271.495435] >> d_hydro shutting down normally D_Hydro [1369413271.495435] >> d_hydro shutting down normally D_Hydro [1369413271.495476] >> d_hydro shutting down normally D_Hydro [1369413271.500990] >> d_hydro shutting down normally ================== ============ Attribute Numerical ================== ============ Tide 40 Wind 41 Humidity 42 Tide 40 Wind 41 Humidity 42 Tide 40 Wind 41 Humidity 42 Tide 40 Wind 41 Humidity 42 Tide 40 ================== ============

CHAPTER 6

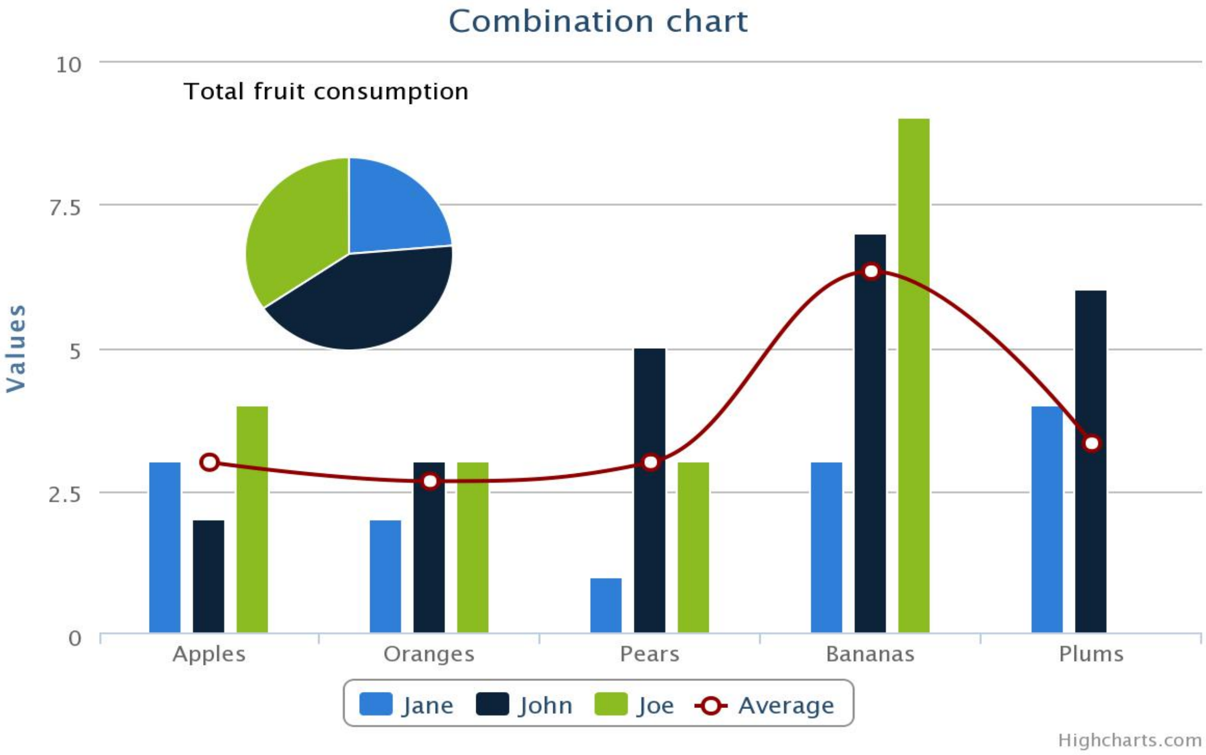
Data

fetch output data and displays it here.

CHAPTER 7

Visualization

Visualize output data, or observal data to make a comparison. sample image:



CHAPTER 8

Discussion

conduct a simple discussion basing on visualization compare.

CHAPTER 9

Sample

9.1 Sphinx Cheat Sheet

Wherein I show by example how to do some things in Sphinx (you can see a literal version of this file below in *This file*)

9.1.1 Making a list

It is easy to make lists in rest

Bullet points

This is a subsection making bullet points

- point A
- point B
- point C

Enumerated points

This is a subsection making numbered points

1. point A
2. point B
3. point C

9.1.2 Making a table

This shows you how to make a table – if you only want to make a list see *Making a list*.

Name	Age
John D Hunter	40
Cast of Thousands	41
And Still More	42

9.1.3 Making links

Cross-references sections and documents

Use reST labels to cross-reference sections and other documents. The mechanism for referencing another reST document or a subsection in any document, including within a document are identical. Place a *reference label* above the section heading, like this:

```
.. _sphinx_helpers:

====================
 Sphinx Cheat Sheet
====================
```

Note the blank line between the *reference label* and the section heading is important!

Then refer to the *reference label* in another document like this:

```
:ref:`sphinx_helpers`
```

The reference is replaced with the section title when Sphinx builds the document while maintaining the linking mechanism. For example, the above reference will appear as *Sphinx Cheat Sheet*. As the documentation grows there are many references to keep track of.

For documents, please use a *reference label* that matches the file name. For sections, please try and make the *refence label* something meaningful and try to keep abbreviations limited. Along these lines, we are using *underscores* for multiple-word *reference labels* instead of hyphens.

Sphinx documentation on Cross-referencing arbitrary locations has more details.

External links

For external links you are likely to use only once, simple include the like in the text. This link to google was made like this:

```
`google <http://www.google.com>`_
```

For external links you will reference frequently, we have created a `links_names.txt` file. These links can then be used throughout the documentation. Links in the `links_names.txt` file are created using the reST reference syntax:

```
.. _targetname: http://www.external_website.org
```

To refer to the reference in a separate reST file, include the `links_names.txt` file and refer to the link through it's target name. For example, put this include at the bottom of your reST document:

```
.. include:: ../links_names.txt
```

and refer to the hyperlink target:

```
blah blah blah targetname_ more blah
```

Links to classes, modules and functions

You can also reference classes, modules, functions, etc that are documented using the sphinx autodoc facilites. For example, see the module `matplotlib.backend_bases` documentation, or the class `LocationEvent`, or the method `mpl_connect()`.

9.1.4 ipython sessions

Michael Droettboom contributed a sphinx extension which does pygments syntax highlighting on ipython sessions

```
In [69]: lines = plot([1,2,3])

In [70]: setp(lines)
  alpha: float
  animated: [True | False]
  antialiased or aa: [True | False]
  ...snip
```

This support is included in this template, but will also be included in a future version of Pygments by default.

9.1.5 Formatting text

You use inline markup to make text *italics*, **bold**, or `monotype`.

You can represent code blocks fairly easily:

```
import numpy as np
x = np.random.rand(12)
```

Or literally include code:

9.1.6 Using math

In sphinx you can include inline math $x \leftarrow y$ $x \forall y$ $x - y$ or display math

$$W^{3\beta}_{\delta_1\rho_1\sigma_2} = U^{3\beta}_{\delta_1\rho_1} + \frac{1}{8\pi 2}\int_{\alpha_2}^{\alpha_2} d\alpha'_2 \left[\frac{U^{2\beta}_{\delta_1\rho_1} - \alpha'_2 U^{1\beta}_{\rho_1\sigma_2}}{U^{0\beta}_{\rho_1\sigma_2}}\right]$$

This documentation framework includes a Sphinx extension, `sphinxext/mathmpl.py`, that uses matplotlib to render math equations when generating HTML, and LaTeX itself when generating a PDF. This can be useful on systems that have matplotlib, but not LaTeX, installed. To use it, add `mathpng` to the list of extensions in `conf.py`.

Current SVN versions of Sphinx now include built-in support for math. There are two flavors:

- pngmath: uses dvipng to render the equation
- jsmath: renders the math in the browser using Javascript

To use these extensions instead, add `sphinx.ext.pngmath` or `sphinx.ext.jsmath` to the list of extensions in `conf.py`.

All three of these options for math are designed to behave in the same way.

9.1.7 Inserting matplotlib plots

Inserting automatically-generated plots is easy. Simply put the script to generate the plot in any directory you want, and refer to it using the `plot` directive. All paths are considered relative to the top-level of the documentation tree. To include the source code for the plot in the document, pass the `include-source` parameter:

```
.. plot:: devel/guidelines/elegant.py
   :include-source:
```

In the HTML version of the document, the plot includes links to the original source code, a high-resolution PNG and a PDF. In the PDF version of the document, the plot is included as a scalable PDF.

9.1.8 Emacs helpers

See *rst_emacs*

9.1.9 Inheritance diagrams

Inheritance diagrams can be inserted directly into the document by providing a list of class or module names to the `inheritance-diagram` directive.

For example:

```
.. inheritance-diagram:: codecs
```

produces:

9.1.10 This file

The computational resources are provided by the Coastal Hazards Research Collaboratory, the Center for Computational and Technology, and the Louisiana Optical Network Initiative. Created using Sphinx 1.1.3.