How to set up a namelist for RAPID

By Cédric H. David (cedric.david@jpl.nasa.gov),
09 Oct 2014, updated 27 Oct 2014, 21 Jan 2015

Background

The Routing Application for Parallel computatIon of Discharge (RAPID) is a river network routing model. The RAPID source code is available at https://github.com/c-h-david/rapid.git and information on RAPID can be found at http://rapid-hub.org/. The original scientific paper for RAPID is available here and the corresponding citation is the following:

David, Cédric H., David R. Maidment, Guo-Yue Niu, Zong-Liang Yang, Florence Habets and Victor Eijkhout (2011), River network routing on the NHDPlus dataset, Journal of Hydrometeorology, 12(5), 913-934. DOI: 10.1175/2011JHM1345.1.

Introduction

This document sheds some light on how to create the text file that contains the instructions read by RAPID at runtime. This text file is called a "namelist" and is required before running RAPID. One of the main advantages of having a namelist is that the various options of RAPID can be changed without recompiling the source code hence enabling multiple simulations with some ease.

Generic name for the RAPID namelist

At runtime, RAPID looks for a generic file called `rapid_namelist` which is located in the following directory: `rapid/run/`. If one has multiple namelists to be used – as is common when running several simulations – one should create a symbolic link (a "shortcut") between a given namelist (e.g. `rapid_namelist_Domain`) and the generic namelist (`rapid_namelist`). In a Unix-like operating system, this can be done using:

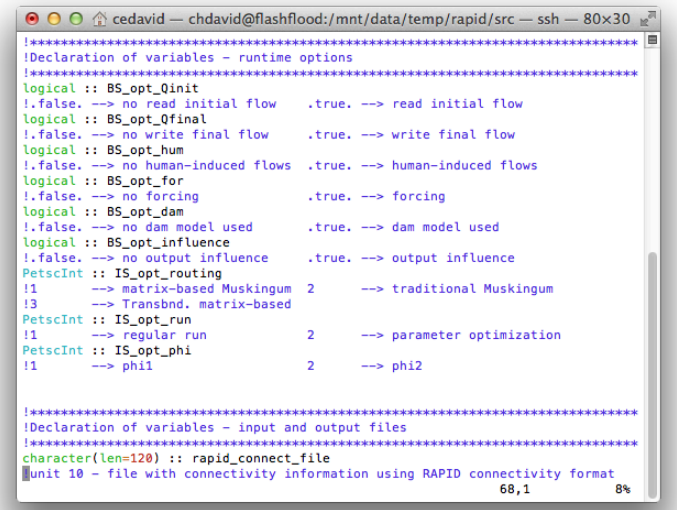
```
cd rapid/run/
ln –s rapid_namelist_Domain rapid_namelist
```

Information about the variables in the RAPID namelist

The RAPID namelist contains many variables and their corresponding values. The first place to look for information on what these variables are is in the comment lines in the following RAPID source code file:

`rapid/src/rapid_var.F90`

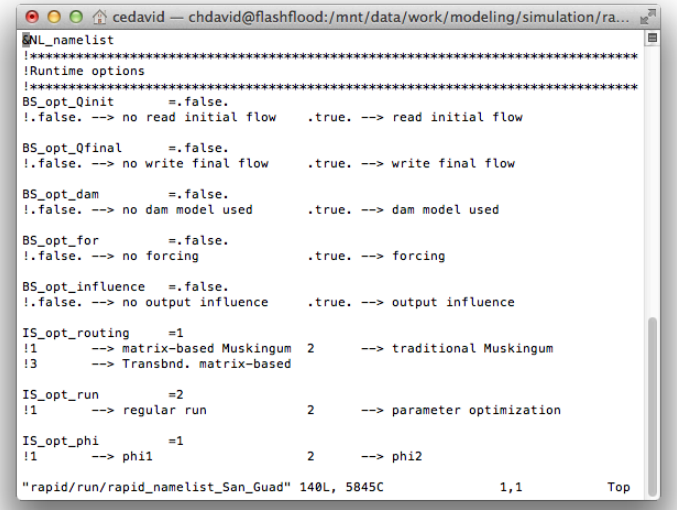
This particular Fortran module is where all the RAPID variables are declared.



Examples and basic structure of the RAPID namelist

Examples of RAPID namelists can be downloaded from the RAPID website at http://rapid-hub.org/download.html.

The first section of the namelist contains high-level options that govern how the model is to run. For example, this first section is where one specifies whether the traditional Muskingum method or the matrix-based Muskingum method is used by RAPID for river routing. This section of the namelist is also where one choses whether to simulate flows (hence generating an output file with flow rates) or optimize model parameters (hence computing a set of multiplying factors to be applied to the existing parameters).



These high-level options are either a logical variable (true or false) or an integer with a limited number of possible values (two or three currently). As of writing, there is a total of 2^7 x 3 = 384 possible combinations of the high-level options. It is therefore challenging to provide a complete description of all possible combinations, and also likely that some have yet to be tested. Further, note that not all variables have to be present in the namelist depending on what combination of options is used at runtime. The best way to understand the dependencies is to look in the source code, or by trial and error.

The remaining sections of the RAPID namelist contain other variable names and values. These include the names and locations of input and output files, the number of river reaches in the computing domain, etc. One should note that RAPID variables all use SI units. Therefore, times are in seconds, volumes in cubic meters, volumetric flow rates in cubic meters per second, etc.

Temporal parameters in the RAPID namelist

Perhaps the least intuitive of the variables in the RAPID namelist are the temporal variables (the first section after high-level options), which are further developed here.

The following temporal variables are only used when running RAPID in regular simulation mode (`IS_opt_run=1`):

- `ZS_TauM` is the duration (in seconds) of the simulation. This duration therefore has to be equal to (or shorter than) the total duration of runoff input data provided in one of the input files (a netCDF file, e.g. `rapid/input/Domain/m3_riv_Domain.nc`).
- `ZS_dtM` is an internal time step (in seconds). This time step corresponds to an internal loop that was initially created to allow for replacement of model computations by observations. This capability is now handled through another temporal variable (`ZS_dtF`, see further down). Therefore, `ZS_dtM` is a legacy variable of which necessity can be challenged, but the RAPID

source code still uses it. Additionally, ZS_dtM is used similarly to another variable (ZS_dtO, see further down) which allows for some consistency in the source code. Because daily observations are common, a value of 86400 seconds (corresponding to 1 day) has traditionally been used. ZS_dtM=86400 still remains the recommended value.

The following temporal variables are only used when running RAPID in optimization mode (IS_opt_run=2):

- ZS_TauO is the duration (in seconds) of each simulation of the optimization procedure. This duration therefore has to be equal to (or shorter than) the total duration of runoff data provided in one of the input files (a netCDF file, e.g. rapid/input/Domain/m3_riv_Domain.nc). Additionally, this duration also has to be equal to (or shorter than) the total duration of the observational data provided in one of the input files (a .csv file, e.g. rapid/input/Domain/Qobs_Domain.csv). RAPID will simulate flows over this duration many times in a given optimization procedure.
- ZS_dtO is an internal time step (in seconds). This time step corresponds to that of the aforementioned observational data.

The following temporal variables are always used when running RAPID:

- ZS_TauR is the duration (in seconds) of the routing procedure. This duration therefore has to be equal to the time step of runoff data provided in one of the input files (a netCDF file, e.g. rapid/input/Domain/m3_riv_Domain.nc).
- ZS_dtR is the routing time step (in seconds). This time depends on the laws of physics and some knowledge of the flow wave celerities and of the river reach lengths in a given river network (the computing domain in RAPID) is needed to pick the value of this variable. Existing studies applying RAPID on the NHDPlus dataset use a value of ZS_dtR=900. Existing studies applying RAPID on the HydroSHEDS dataset use a value of ZS_dtR=1800. The RAPID source code assumes that ZS_dtR is smaller than ZS_tauR.

The following temporal variable is only used in "forcing mode", i.e. when replacing some of the RAPID flow computations by observations (BS_opt_for=.true.) at runtime:

- ZS_dtF is the forcing time step (in seconds) at which forcing data replaces RAPID flow computations. Therefore, this time step has to be the same time step at which forcing data provided in one of the input files is available (a .csv file, e.g. rapid/input/Domain/Qfor_Domain.csv).

One final note: the following ratios must be exact integers:

- ZS_TauM/ZS_dtM, ZS_TauO/ZS_dtO, and ZS_TauR/ZS_dtR.
- ZS_dtM/ZS_TauR, ZS_dtO/ZS_TauR, and ZS_dtF/ZS_TauR.

Further information

RAPID website: http://rapid-hub.org/

RAPID source code: https://github.com/c-h-david/rapid/