RAPID data model

By Cédric H. David (cedric.david@jpl.nasa.gov),
06 Jul 2010, updated 21 Jan 2015

Goal

Given the following information on a river network domain:

- Network connectivity
- Inflow to river network from land and aquifers (provided by a land surface model for example)
- Estimate of model parameters (can be crude estimate)
- Stream flow observations at many stations
- Forcing data, i.e. flow data that is to be used as upstream flow instead of upstream flow computed by RAPID (forcing can be from observations)

Given a list of IDs of rivers constituting a basin that is fully contained in the domain (can be the domain itself)

RAPID can:

- Compute Q and V for the basin
- Optimize Muskingum k and x for basin based on observed flows

Underlying assumptions

Stream flow observations are from gage measurements. A subset of all gages available in the domain can be used in the optimization procedure.

Forcing data is a term used loosely here to indicate that known flow coming for upstream is used at given locations.

© 2015 Cédric H. David
All Rights Reserved

Input files in RAPID

RAPID has both input and output (I/O) files. In Fortran, I/O files have to have a unit number. For clarity of the code, RAPID uses the same unit numbers throughout all subroutines.

10 rapid_connect_file
11 riv_bas_id_file
12 obs_tot_id_file
13 obs_use_id_file
16 for_tot_id_file
17 for_use_id_file

20 k_file
21 x_file
22 kfac_file
23 xfac_file

30 Qinit_file
31 Qfinal_file
xx Vlat_file
33 Qobs_file
34 Qfor_file
35 Qobsbarrec_file

xx Qout_file

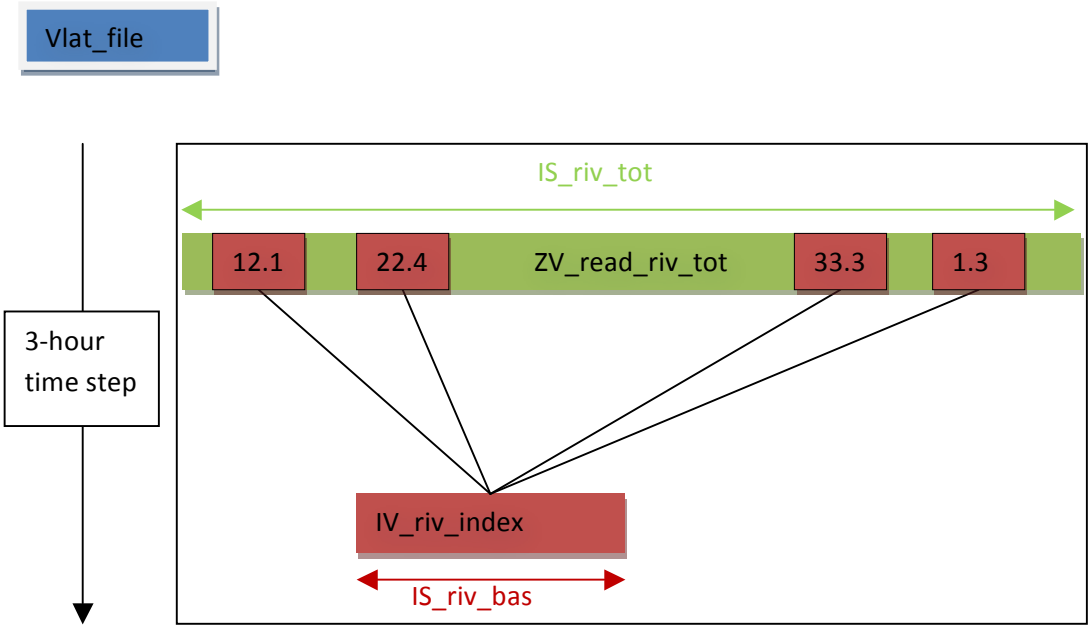
Sorting within RAPID files

The ordering of reach IDs in the corresponding values of variables in all following files has to be consistent with rapid_connect_file: kfac, Qinit, k, x, Vlat. This is because the variable IV_riv_index is used when reading all these files and IV_riv_index is calculated based on riv_bas_id_file and rapid_connect_file.

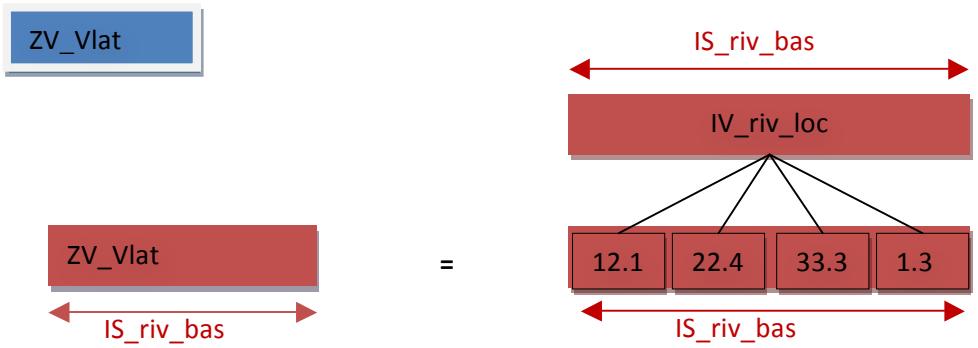
However, the order of riv_bas_id_file doesn't matter for reading the inputs. It only matters for the calculations within RAPID and hence for performance of the linear system solvers.

© 2015 Cédric H. David
All Rights Reserved

Reading lateral inflow



Read → ZV_read_riv_tot



IS_riv_tot is known directly from the size of domain, it is also the number of lines in rapid_connect_file

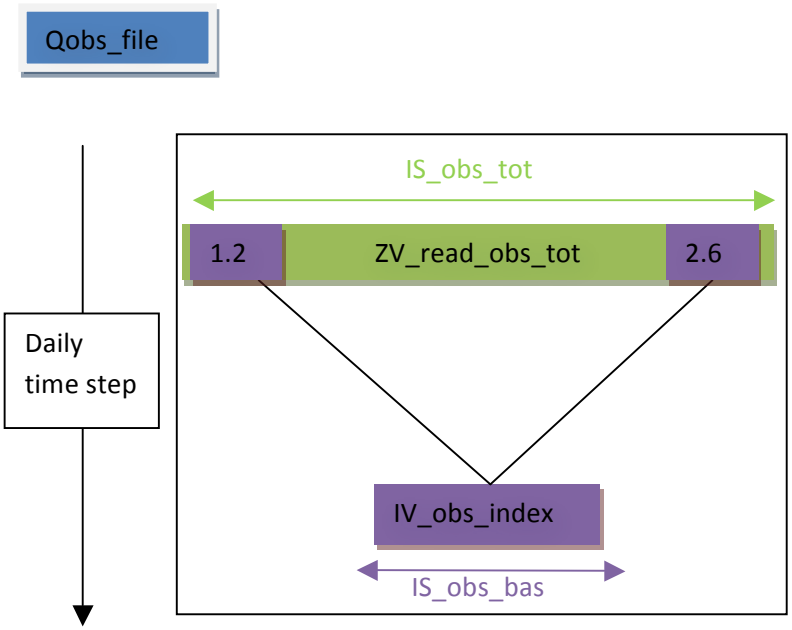
IS_riv_bas is known directly from the size of basin, it is also the number of lines in riv_bas_id_file

Within PETSc, the way to do "ZV_Vlat(IV_riv_bas_loc)=ZV_read_riv_tot(IV_riv_index)" is:

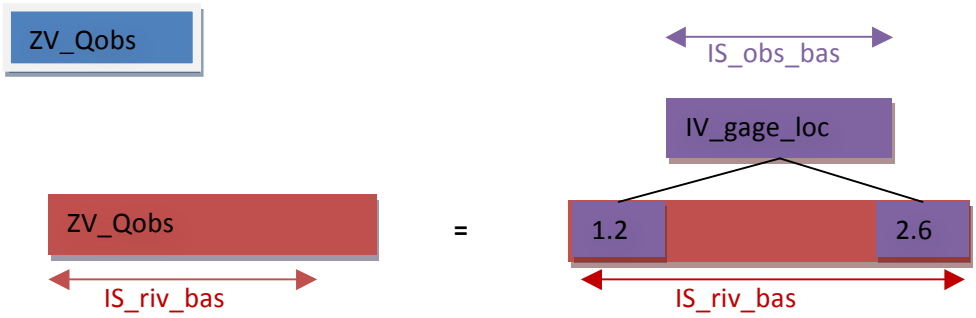
VecSetValues(ZV_Vlat,IS_riv_bas,IV_riv_bas_loc,ZV_read_riv_tot(IV_riv_index),ierr)

© 2015 Cédric H. David
All Rights Reserved

Reading observations



Read → ZV_read_obs_tot



IS_obs_tot the total number of gages in domain, it is also the size of obs_tot_id_file.

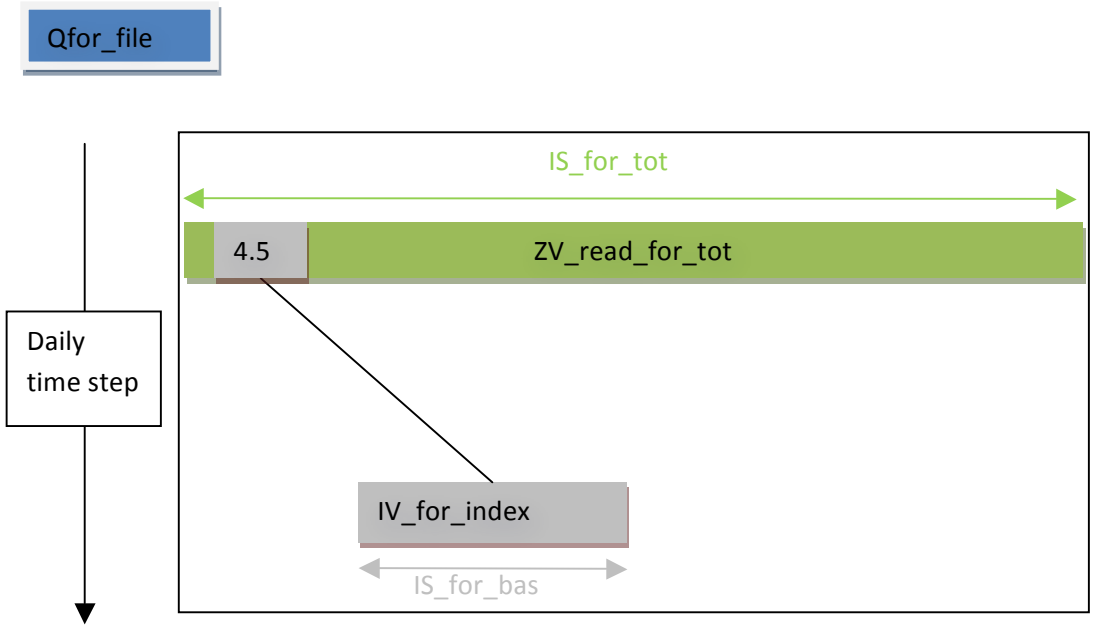
IS_obs_bas is determined on the fly based on obs_use_id_file and riv_bas_id_file, therefore of IV_obs_index and IV_gage_loc have to be allocated within code.

Within PETSc, the way to do "ZV_Qobs(IV_riv_bas_loc)=ZV_read_obs_tot(IV_obs_index)" is:

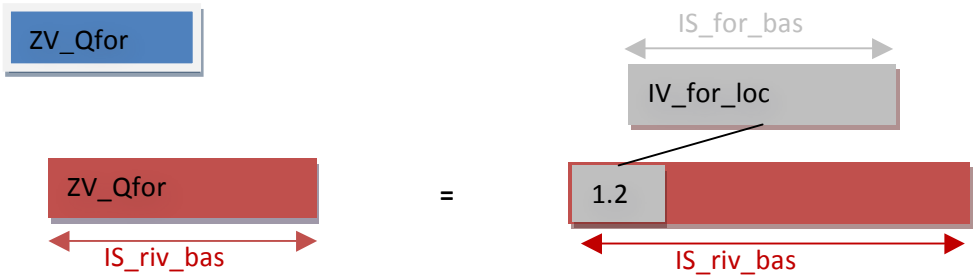
VecSetValues(ZV_Qobs,IS_obs_bas,IV_gage_loc,ZV_read_obs_tot(IV_obs_index),ierr):

© 2015 Cédric H. David
All Rights Reserved

Reading of forced inflow



Read → ZV_read_for_tot



IS_for_tot is the total number of available forcing locations, it is also the size of for_tot_id_file

IS_for_bas is determined on the fly based on for_use_id_file, riv_bas_id_file, and rapid_connect_file. Therefore of IV_for_index and IV_for_loc would have to be allocated within code. *The trick here is that forcing is not applied at the reach where data is measured, it's the reach downstream of measured data.*

VecSetValues(ZV_Qfor,IS_for_bas,IV_for_loc,ZV_read_for_tot(IV_for_index),ierr):

ZV_Qfor(IV_for_loc)=ZV_read_for_tot(IV_for_index)

© 2015 Cédric H. David
All Rights Reserved

Further information

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

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

© 2015 Cédric H. David
All Rights Reserved