Input Data Formats

- **Object type based**
  - Rather than subbasin & HRU based
  - No. input files remain the same regardless of No. subbasins & HRUs
    - Easy to handle input files
    - Efficient for large & fine-scale modeling & calibration
      - 137 Subbasins & 1212 HRUs (Brentwood watershed)
      - 40 files (modular) vs. 9400 files (rev. 629)

- **Can we still use input files in the old formats?**

Fig.1. Converter from old input file formats to the new ones (Developed by Dr. White)

From “ProjectName.dbf” to a set of input files in the new format
Input Data Formats

• How relationships between objects are defined?

**Fig. 1. Example of input data format (‘subbasin.con’)**

<table>
<thead>
<tr>
<th>Subbasin No.</th>
<th>Property ID</th>
<th>Total number of outgoing (source) objects</th>
<th>Type &amp; ID of objects to which routed</th>
<th>Outflow hydrograph type &amp; fraction</th>
<th>OBSTYP_OUT</th>
<th>OBSTYPNO_OUT</th>
<th>HTRP_OUT</th>
<th>FRAC_OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>sub001</td>
<td>1</td>
<td>cha</td>
<td>COT</td>
<td>1</td>
<td>1</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>sub002</td>
<td>2</td>
<td>cha</td>
<td>COT</td>
<td>1</td>
<td>1</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>sub003</td>
<td>3</td>
<td>cha</td>
<td>COT</td>
<td>1</td>
<td>1</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>sub004</td>
<td>4</td>
<td>cha</td>
<td>COT</td>
<td>1</td>
<td>1</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>sub005</td>
<td>5</td>
<td>cha</td>
<td>COT</td>
<td>1</td>
<td>1</td>
<td>1.0</td>
<td></td>
</tr>
</tbody>
</table>

**Fig. 2. Example of input data format (‘channel.con’: similar function to that of ‘fig.fig’)**

<table>
<thead>
<tr>
<th>Subbasin No.</th>
<th>Weather Station ID</th>
<th>Cumulative drainage area (ha)</th>
<th>Channel ID</th>
<th>Total number of outgoing (source) objects</th>
<th>ID of outgoing (source) objects</th>
<th>OBSTYP_OUT</th>
<th>OBSTYPNO_OUT</th>
<th>HTRP_OUT</th>
<th>FRAC_OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1.1</td>
<td>cha</td>
<td>1</td>
<td>cha</td>
<td>1</td>
<td>1</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1.3</td>
<td>cha</td>
<td>1</td>
<td>cha</td>
<td>1</td>
<td>1</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1.1</td>
<td>cha</td>
<td>1</td>
<td>cha</td>
<td>1</td>
<td>1</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1.1</td>
<td>cha</td>
<td>1</td>
<td>cha</td>
<td>1</td>
<td>1</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>2.7</td>
<td>cha</td>
<td>1</td>
<td>cha</td>
<td>1</td>
<td>1</td>
<td>1.0</td>
<td></td>
</tr>
</tbody>
</table>
Model Structure

- Flexible modeling framework
  - HRU to HRU routing is possible
    - Direct runoff generated in a HRU can go any HRU
    - HRU-based overland (landscape) routing
  - Easily add objects & modify connections between objects
    - Point sources, irrigation, Karst, terraced paddy fields, etc.
Model Structure

• Overall structure

Load Modules

'parm'
'hydrograph_module'
'subbasin_module'
'hru_module'
'wateruse_module'
'climate_module'
'aquifer_module'
'channel_module'
'basin_module'
'jrw_datalib_module'
'reservoir_module'

etc.

Read Inputs

Simulation settings, properties of objects, connection between objects, etc.

'main'

'time_control'

Daily Simulation

'command'

Calculation by objects

• Incorporation of Management Practices
• Compute biological mixing at the end of each year

Object Control

'hru_control'
'subbasin_control'
'aqu_1dlag'
'channel_control'
'res_control'

• Recall daily, monthly & annual hydrographs
• Print all outflow hydrographs

*** Spatial object (sp_ob) vs. object (ob;) ***
• 'sp_ob': No. each type of object
• 'sp_ob1': Starting index of each object in a full list of objects
• i.e. 5 HRUs, 3 subbasins, 3 aquifer, and 3 channels
  • sp_ob%hru = 5, sp_ob1%hru = 1, sp_ob%sub = 3, sp_ob1%sub = 6, sp_ob%aqu = 3, sp_ob1%aqu = 9, sp_ob%cha = 3, and sp_ob1%cha = 12
  • ob(1) to ob(5) mean “hrus”, ob(6) to ob(8) represent subbasins, and so on
Model Structure

- Data Type?

```fortran
  type object_connectivity
    character(len=16) :: name = "hru_default"
    integer :: typ = 1  ! object type - ie hru, hru_lte, sub, chan, res, recall
    integer :: props = 1  ! properties number from database (ie hru.dat, sub.dat)
    integer :: props2 = 1  ! additional properties number from database (ie chan_weather)
    integer :: wst = 1  ! weather station number
    real :: ha = 0.0  ! need to be changed to read areas from input files
    integer :: fired = 0  ! 0=not fired; 1=fired off as a command
    integer :: cmd_next = 0  ! next command (object) number
    integer :: cmd_prev = 0  ! previous command (object) number
    integer :: cmd_order = 0  ! 1=headwater, 2=2nd order, etc
    integer :: src_tot = 0  ! total number of outgoing (source) objects
    integer :: rcv_tot = 0  ! total number of incoming (receiving) hydrographs
    integer :: rcvob_tot = 0  ! total number of incoming (receiving) objects
    integer :: dfn_tot = 0  ! total number of defining objects (ie hru's within a subbasin)
    integer :: subs_tot = 0  ! number of subbasins that contain this object
    integer :: elem = 0  ! subbasins element number for this object
    integer :: flood_ch_nb ! channel that landscape unit is linked to
  character(len=3), dimension(:,), allocatable :: obtyp_out ! outflow object type (ie 1-hru, 2=sd_hru, 3=3, 4-chan, etc)
  integer, dimension(:,), allocatable :: obtypn_out ! outflow object type name
  integer, dimension(:,), allocatable :: ob_out ! outflow object
  character(len=3), dimension(:,), allocatable :: hytyp_out ! outflow hyd type (ie 1-tot, 2= recharge, 3=surf, etc)
  integer, dimension(:,), allocatable :: htyp_out ! outflow hyd type (ie 1-tot, 2= recharge, 3=surf, etc)
  real, dimension(:,), allocatable :: frac_out ! fraction of hydrograph
  integer, dimension(:,), allocatable :: obtyp_in ! inflow object type (ie 1-hru, 2=sd_hru, 3=3, 4-chan, etc)
  integer, dimension(:,), allocatable :: obtypn_in ! inflow object type name
  integer, dimension(:,), allocatable :: obj_in ! inflow object
  integer, dimension(:,), allocatable :: htyp_in ! inflow hyd type (ie 1-tot, 2= recharge, 3=surf, etc)
  real, dimension(:,), allocatable :: frac_in ! inflow hydrograph for surface runon - sum of all inflow hyds
  type (hyd_output), dimension(:,), allocatable :: hinf  ! generated hydrograph (ie 1-tot, 2= recharge, 3=surf, etc)
  real, dimension(:,), allocatable :: ts ! peak flow rate during time step - m3/s
  type (hyd_output), dimension(:,), allocatable :: hinf_m
  type (hyd_output), dimension(:,), allocatable :: hinf_y
  type (hyd_output), dimension(:,), allocatable :: hinf_n
  type (hyd_output), dimension(:,), allocatable :: hout_m
  type (hyd_output), dimension(:,), allocatable :: hout_y
  type (hyd_output), dimension(:,), allocatable :: hout_n
  type (hyd_output), dimension(:,), allocatable :: hdcp_m
  type (hyd_output), dimension(:,), allocatable :: hdep_m
  integer, dimension(:,), allocatable :: obj_sub # subbasins object number that contain this object
end type object_connectivity
```

- `i`: object index
- ‘subs_tot’: a property of `ob(i)`
Challenges

• Runtime Initialization
  – Over 30 seconds with 137 subbasins & 1212 HRUs
  – Parallel loading?

• Routing scheme between spatial objects
  – Hard to track variables of each object
  – User Manual needed for developers

• Parallelization
  – HRU and subbasin-level computations are independent
  – Channel routing is inherently hierarchical process
    • But there are some options (P-SWAT, Wu et al., 2012)
SWAT for Brentwood Watershed

- **Brentwood WS**
  - Austin, TX
  - 149.8 ha
  - Highly urbanized
  - Monitored by City of Austin

- **SWAT**
  - Prepared by City of Austin
  - Great details
    - 137 subbasins (1.1 ha/sub)
    - 1212 HRUs (0.12 ha/HRU)
  - Calibrated by BRC
SWAT for Brentwood Watershed

- Calibrated SWAT
  - ‘Good’ performance; overestimated runoff volume

Fig. 1. Comparison of observed & simulated daily runoff

Fig. 2. Comparison of observed & simulated monthly runoff hydrographs

Table. 1. Performance statistics of the calibrated SWAT model

<table>
<thead>
<tr>
<th>Period</th>
<th>NSE</th>
<th>R²</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>15-min</td>
<td>Daily</td>
</tr>
<tr>
<td>Calibration</td>
<td>0.88</td>
<td>0.91</td>
</tr>
<tr>
<td>Validation</td>
<td>0.71</td>
<td>0.84</td>
</tr>
</tbody>
</table>