Advanced ParFlow Short Course
CLM Activities

October 3-4, 2019

University of Arizona

Laura Condon, University of Arizona
Nick Engdahl, Washington State University
Reed Maxwell, Colorado school of Mines

This material and short course was supported by the National Science Foundation and Department of Energy

Common Land Model

Activity 1: Finding Variables

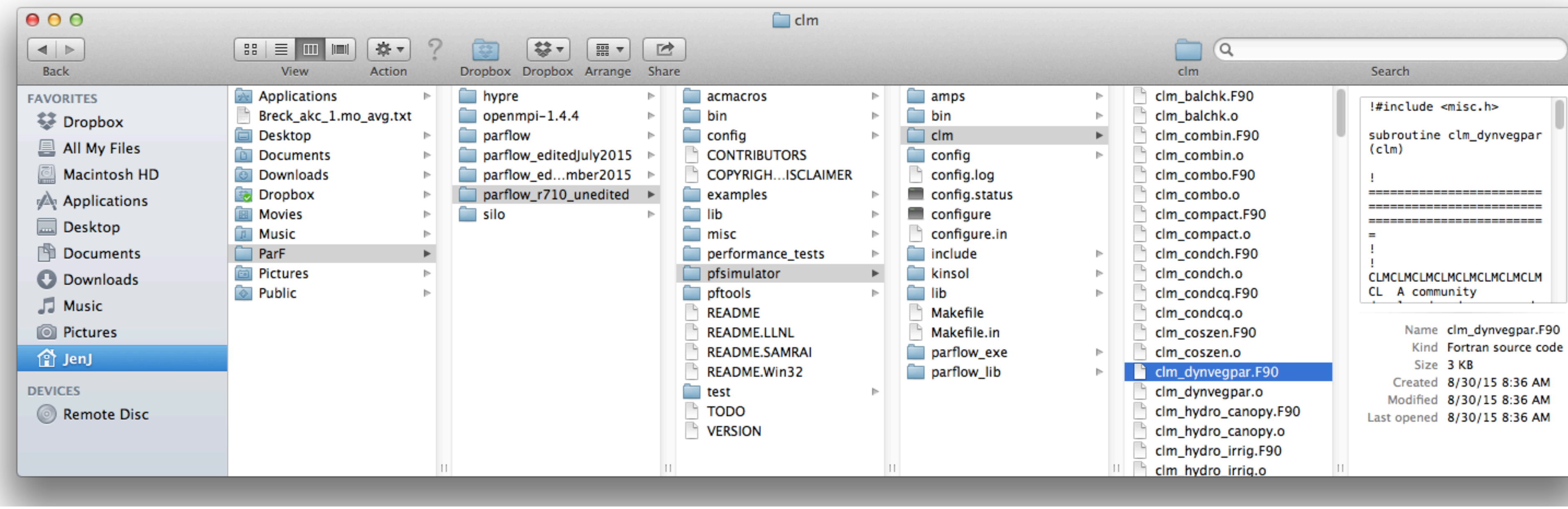
Developed by:
Jennifer Jefferson

Why should you know how to find variables in the CLM code?

- Inevitably, someone will ask: “How does the model compute *x*?”
- You want to change how the model computes *x*
- You want to know where a variable from the input file is used

Where do you search for variables?

pfsimulator – clm folder includes all CLM .F90 modules
parflow_lib folder includes solver_richards.c



How do you search for a variable?

(on a Mac)

- Type in name to search bar in Finder window
- Use `grep` (global regular expression print) command from terminal window

Variations of `grep`

- http://www.gnu.org/software/grep/manual/grep.htm

- -l list names of files
- -r recursively look through files in a given directory
- * wildcard symbol

- `grep` command is very similar on Linux
  http://www.tecmint.com/12-practical-examples-of-linux-grep-command/

  http://www.computerhope.com/unix/ugrep.htm

Let’s do an example:

sand

(from drv_vegm.dat)

1. Navigate to director to search in

2. Search for “sand” in Fortan files within the directory

```
clm — bash — 129×24
bash                                   bash
Last login: Tue May 17 09:51:53 on ttys000
Sonyas-MacBook-Air:pfsimulator JenJ$ cd /Users/JenJ/ParF/parflow_r710_unedited/pfsimulator/clm
Sonyas-MacBook-Air:clm JenJ$ grep "sand" *.F90
drv_clmini.F90:!@      clm%watsat(j) = 0.489 - 0.00126*tile%sand(j)*100.0 Stefan: followed Reed to make it consistent with PILPS
drv_clmini.F90:        xksat         = 0.0070556 *( 10.**(-0.884+0.0153*tile%sand(j)*100.0))
drv_clmini.F90:        clm%sucsat(j) = 10. * ( 10.**(1.88-0.0131*tile%sand(j)*100.0) )
drv_clmini.F90:        tkm           = (8.80*tile%sand(j)*100.0+2.92*tile%clay(j)*100.0) /  &
drv_clmini.F90:                        (tile%sand(j)*100.0+tile%clay(j)*100.0)          ! W/(m K)
drv_clmini.F90:        clm%csol(j)   = (2.128*tile%sand(j)*100.0+2.385*tile%clay(j)*100.0)/ &
drv_clmini.F90:                        (tile%sand(j)*100.0+tile%clay(j)*100.0)*1.e6     ! J/(m3 K)
drv_gridmodule.F90:     real(r8) :: sand(nlevsoi) ! Percent sand in soil
drv_pout.F90:     write(n,5)'Percent SAND               lay: ', l, drv_gridave(nch,mask,tile%fgrd,tile%sand(l), drv)
drv_readvegtf.F90:  real(r8) :: sand          !temporary value of input sand
drv_readvegtf.F90:                sand,              &
drv_readvegtf.F90:       grid(c-ix,r-iy)%sand(:) = sand
drv_readvegtf.F90:                tile(drv%nch)%sand(:) = grid(c,r)%sand(:)   !Percent sand in soil
drv_tilemodule.F90:     real(r8) :: sand(nlevsoi) ! Percent sand in soil (vertically average)
Sonyas-MacBook-Air:clm JenJ$
```

Example (sand)

```
clm — bash — 119×24
bash | bash | bash
Last login: Tue May 17 14:21:23 on ttys001
Sonyas-MacBook-Air:clm JenJ$ cd /Users/JenJ/ParF/parflow_r710_unedited/pfsimulator/clm
Sonyas-MacBook-Air:clm JenJ$ grep "csol" *.F90
clm_main.F90:  !     csol,watsat,sucsat,bsw,tkmg,tksatu,tkdry,hksat,wtfact,trsmx0
clm_thermalk.F90:          cv(i) = clm%csol(i)*(1-clm%watsat(i))*clm%dz(i) +   &
clm_typini.F90:     clm(k)%csol  (:) = NaN      ! heat capacity, soil solids (J/m**3/Kelvin)
clmtype.F90:       real(r8) :: csol  (nlevsoi) ! heat capacity, soil solids (J/m**3/Kelvin)
drv_clmini.F90:         clm%csol(j)   = (2.128*tile%sand(j)*100.0+2.385*tile%clay(j)*100.0)/ &
drv_clmini.F90:         clm%csol(j)   = drv%udef
drv_pout.F90:      write(n,5)'Heat cap of soil soilds     lay: ', l, drv_gridave(nch,mask,tile%fgrd,clm%csol(l), drv)
Sonyas-MacBook-Air:clm JenJ$ grep "tkmg" *.F90
clm.F90:               clm(t)%tksatu(k)        = clm(t)%tkmg(k)*0.57**clm(t)%watsat(k)
clm_main.F90:  !     csol,watsat,sucsat,bsw,tkmg,tksatu,tkdry,hksat,wtfact,trsmx0
clm_thermalk.F90:                dksat = clm%tkmg(i)*0.249**(fl*clm%watsat(i))*2.29**clm%watsat(i)
clm_typini.F90:     clm(k)%tkmg  (:) = NaN      ! thermal conductivity, soil minerals  [W/m-K]
clmtype.F90:       real(r8) :: tkmg  (nlevsoi) ! thermal conductivity, soil minerals  [W/m-K]
drv_clmini.F90:         clm%tkmg(j)   = tkm ** (1.- clm%watsat(j))
drv_clmini.F90:         clm%tksatu(j) = clm%tkmg(j)*0.57**clm%watsat(j)
drv_clmini.F90:         clm%tkmg(j)   = drv%udef
drv_pout.F90:      write(n,5)'Thermal conduct of soil min lay: ', l, drv_gridave(nch,mask,tile%fgrd,clm%tkmg(l), drv)
Sonyas-MacBook-Air:clm JenJ$
```

Example (sand)

Variable Name	sand
Description	percent sand in soil
Units	decimal (-)
Global or Local variable	global
Constant or varying?	Constant for each tile, can vary between tiles
CLM Modules	drv_clmini.F90 drv_gridmodule.F90 drv_pout.F90 drv_readvegtf.F90 drv_tilemodule.F90
If/Do statements?	For every soil layer (nlevsoi)
Additional comments	Sand used to compute variables clm%tkmg, clm%csol -> clm_thermalk.F90 -> heat capacity for ground heat flux

You do: Pick variable(s) you might be interested in, use `grep` command to find out more information

Vegetation	Ground	Atmosphere
efpot	eflx_evap_soi	forc_hgt_t
eflx_evap_tot	surfalb	hsub
fdry	zsno	displa
qflx_through	csoilc	co2, pco2
seasb	t_soisno	forc_pbot

You do: Find information about variable(s)

Variable Name	
Description	
Units	
Global or Local variable	
Constant or varying?	
CLM Modules	
If/Do statements?	
Additional comments	

You do: Find information about variable(s)

Variable Name	
Description	
Units	
Global or Local variable	
Constant or varying?	
CLM Modules	
If/Do statements?	
Additional comments	

Tips

- Make sure lines are not commented out (!)
- Code updates are typically annotated with developers initials, for example:
  - RMM = Reed Maxwell
  - IMF = Ian Ferguson
  - NBE = Nick Engdahl
- Be careful of rabbit holes - some variables are more difficult to track down and follow than others
  - Find where variable is initialized
  - Look at flowchart to see order modules are called

Common Land Model

Activity 2: Testing Single Columns

Developed by:
Jennifer Jefferson

Note: Demo is for operation of PF-CLM from a local version installed on a Mac.

Why would you want to test CLM with single columns?

- Computationally cheaper than whole domain simulation
- Can quickly run simulation on laptop
- Isolate variable
- Obtain information about variable not output (by default) from CLM
- Isolate period of time or scenario
- Compare effect of change
- Clarify computation or conceptual understanding

Activity 2

(demo portion)

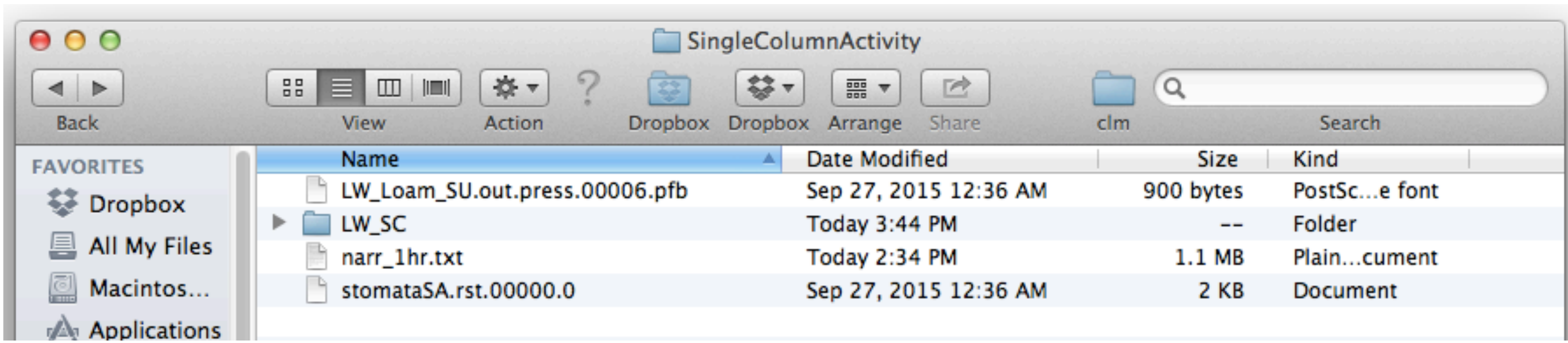
Goal: Output and plot 2 variables from CLM

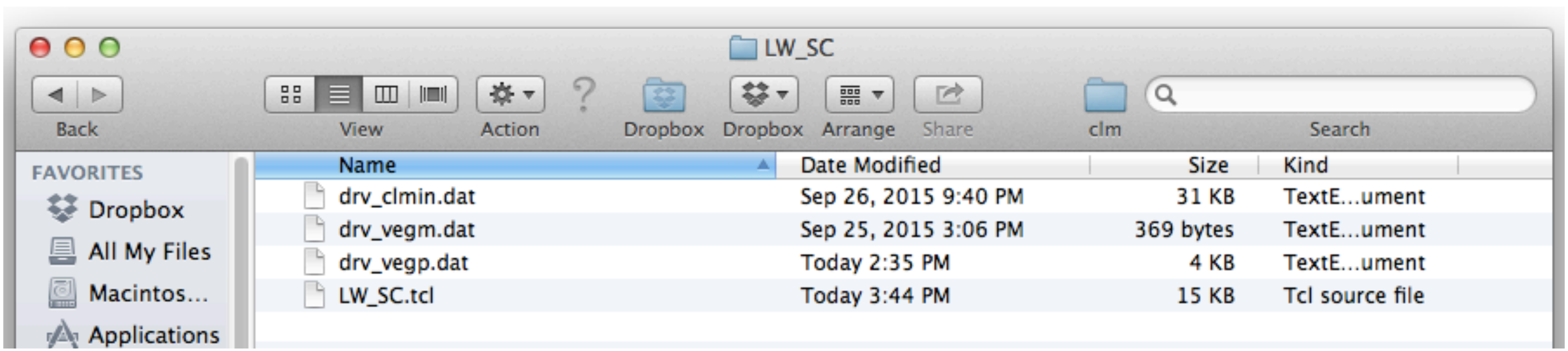
Variables of interest:

1. clm%btran – vegetation water stress (for transpiration)
2. taf – canopy air temperature

- Both variables are located in clm_leaftem.F90
- Neither variable is written out of the model

Step 1: Obtain (or locate) files in a folder called "SingleColumnActivity"

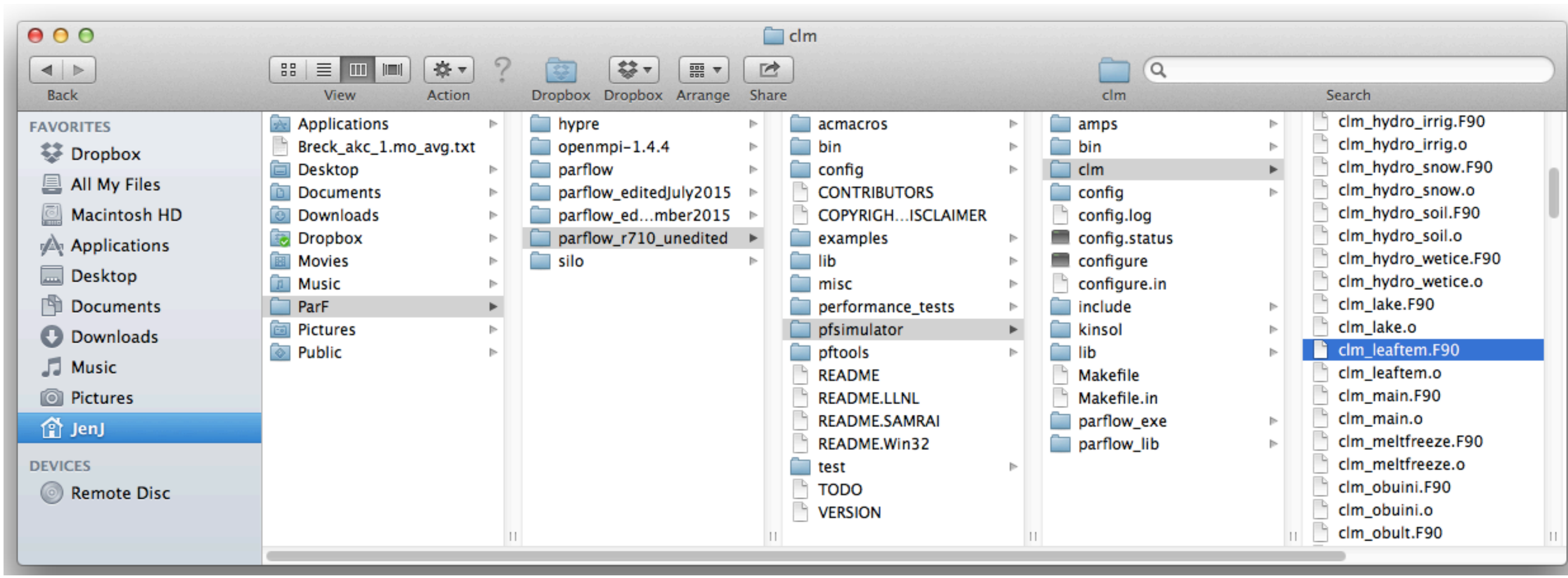




Put folder somewhere on your computer, but leave files in this arrangement.

Step 2: Add print statement to clm_leaftem.F90

- Navigate to clm_leaftem.F90



- Open the file and at the bottom of clm_leaftem.F90 add (will print to ...out.txt):

```
! Update dew accumulation (kg/m2)

clm%h2ocan = max(dble(0.),clm%h2ocan + (clm%qflx_tran_veg-clm%qflx_evap_veg)*clm%dtime)

print*,'111', clm%btran, taf

end subroutine clm_leaftem
```

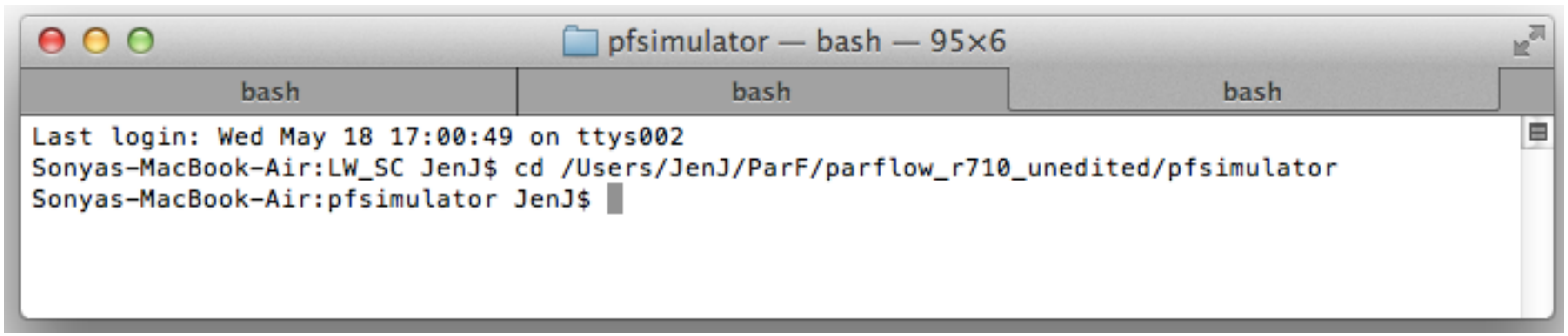
Tips

- Be thoughtful about placement of print statement
  - Not in a loop
  - Before final value is computed
- Select variables strategically so that you can make offline calculations
  - To duplicate output
  - Think ahead, if possible

Step 3: Recompile PF-CLM

- Open the terminal window
- Navigate to pfsimulator folder

  `cd ...`



- Recompile code by typing

  `make install`

Step 4: Run tcl script

- Open a new tab in the terminal window (command+t)
- Navigate to the “SingleColumnActivity – LW_SC” folder

```
cd ...
```

- Run tcl script

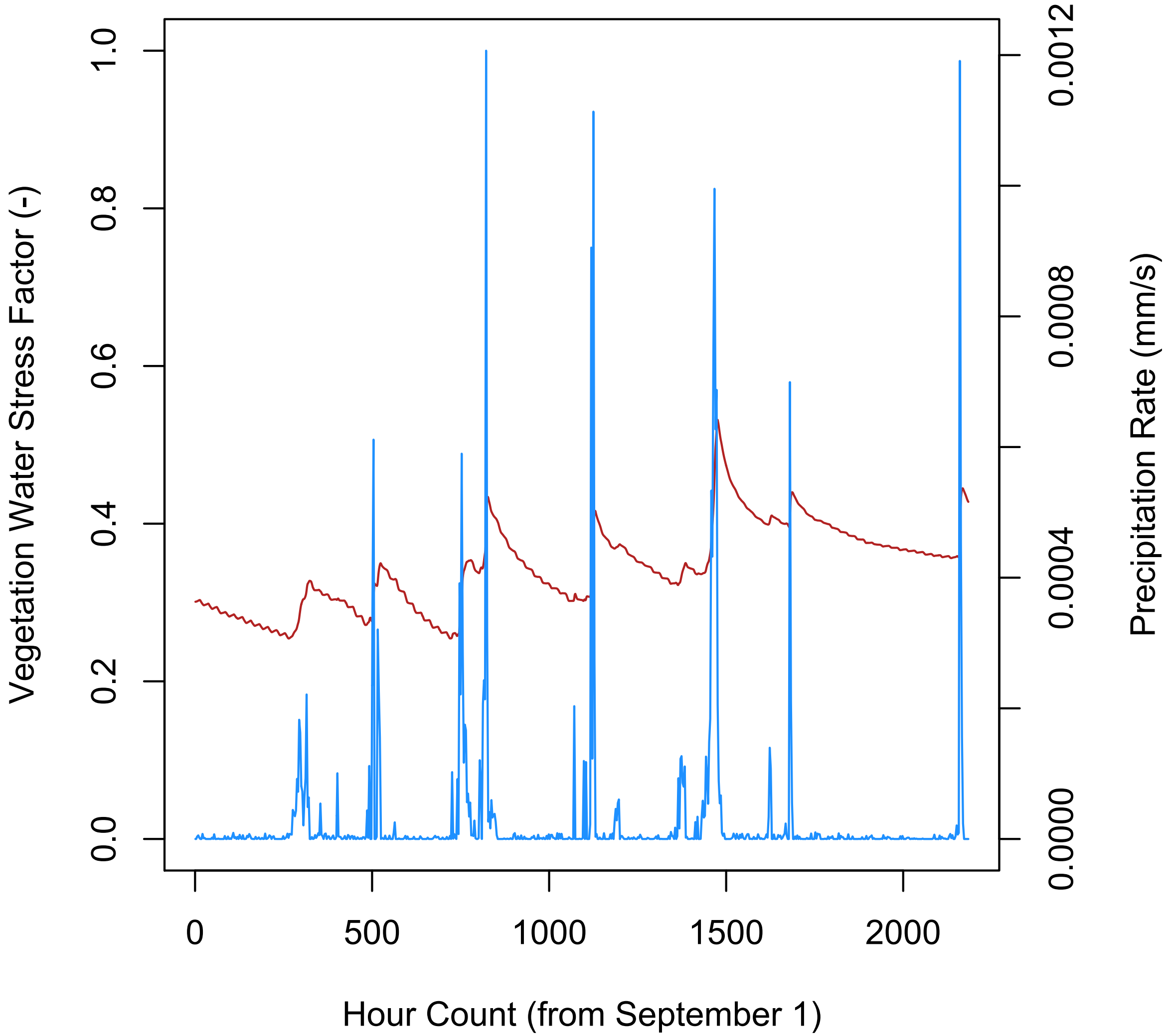
```
tclsh LW_SC.tcl
```

Step 5: Post-process data

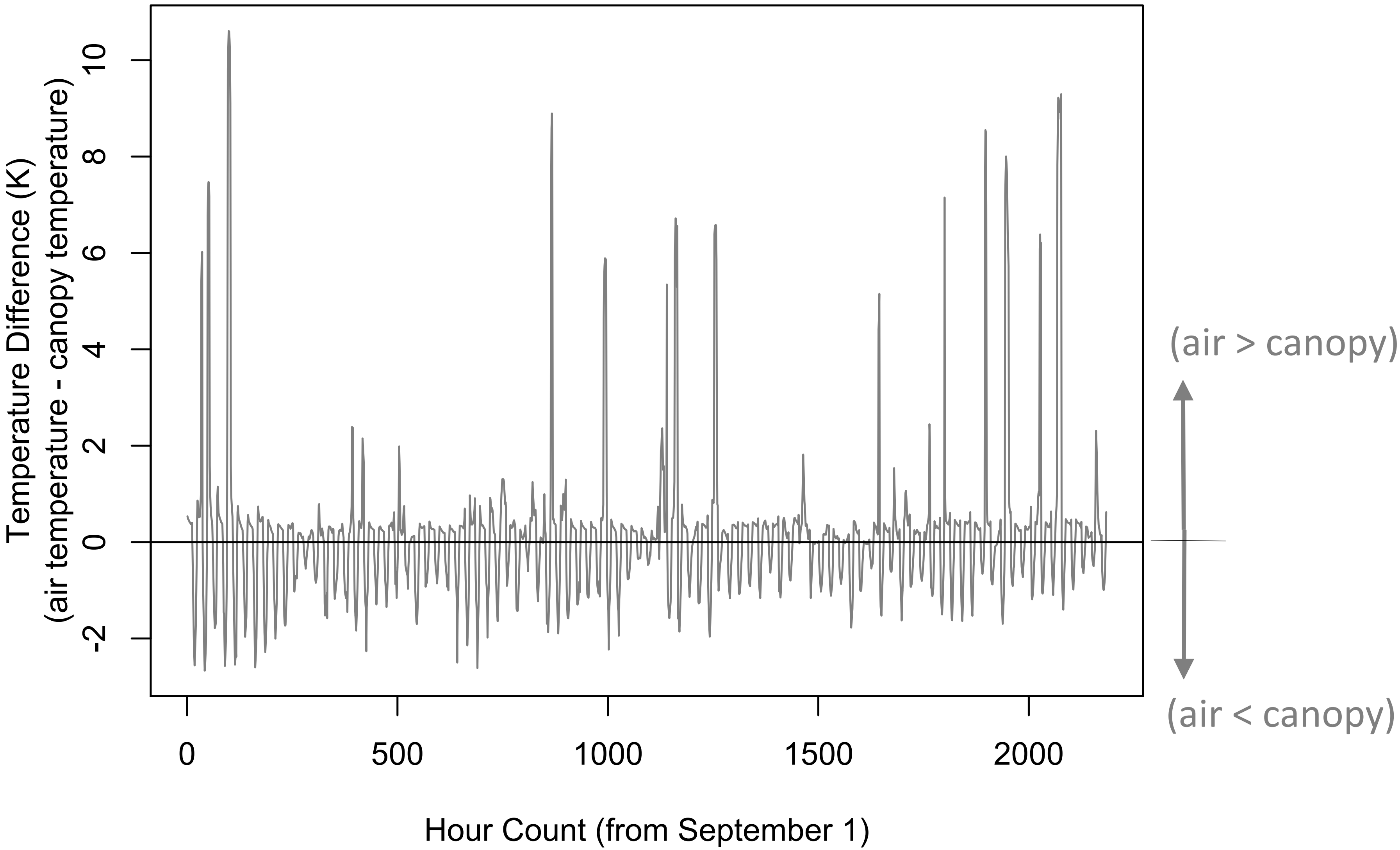
(I used R to load, compare and plot data)

(more moisture, does not limit photosynthesis as much)

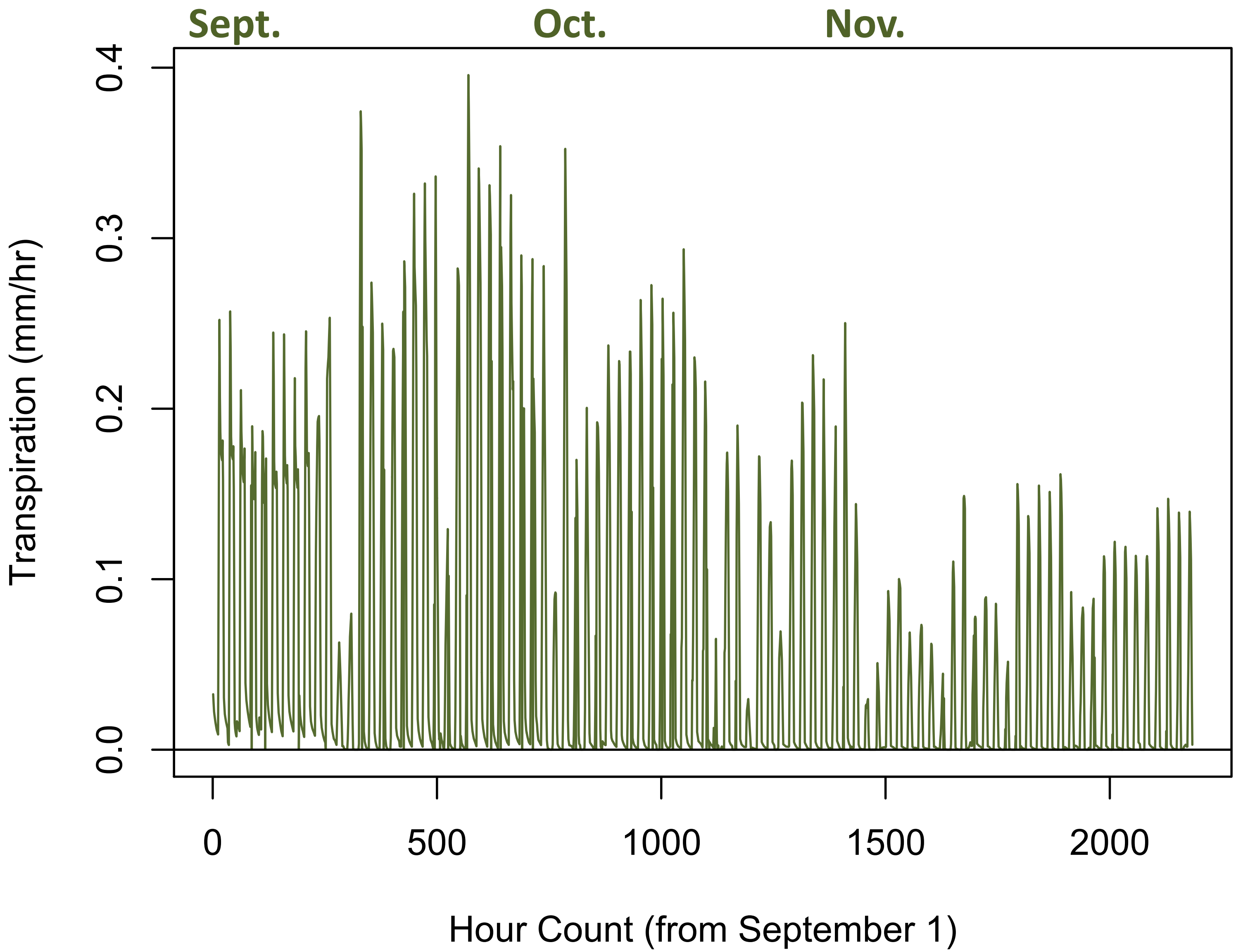
(less moisture, limits photosynthesis more)



Step 5: Post-process data



Step 5: Post-process data



Activity 2

(active portion)

Goal: Run a single column domain and plot variable from single file output

Example single column setup is provided:

1. .tcl script
2. (3) CLM files
3. 1D forcing file
   - Little Washita, OK
   - File begins on September 1, 1998 at 0 GMT (7pm CT August 31, 1998)
4. Pressure file
5. Restart file

(Some) Decisions to Make

1. How many time steps to simulate?

```
set stopt 2184.0

pfset TimingInfo.BaseUnit           1.0
pfset TimingInfo.StartCount         0
pfset TimingInfo.StartTime          0.0
pfset TimingInfo.StopTime           $stopt
pfset TimingInfo.DumpInterval       1.0
pfset TimeStep.Type                      Constant
pfset TimeStep.Value                     1.0
```

2. Restart or not?

```
#pfset ICPressure.Type                                 HydroStaticPatch
#pfset ICPressure.GeomNames                            domain
#pfset Geom.domain.ICPressure.Value                    -1.0
#pfset Geom.domain.ICPressure.RefGeom                  domain
#pfset Geom.domain.ICPressure.RefPatch                 z-upper

pfset ICPressure.Type                                 PFBFile
pfset ICPressure.GeomNames                            domain
pfset Geom.domain.ICPressure.FileName                 "LW_Loam_SU.out.press.00006.pfb"
pfdist "LW_Loam_SU.out.press.00006.pfb"
```

1. What CLM variable to plot?
   - See end of introduction slides for list and order