Introduction to ParFlow-CLM

How it is different from ParFlow and how to use it

ParFlow Short Course

Module 7

Learning Objectives:

At the end of this module students will understand:

- What are the capabilities that CLM adds when coupled with ParFlow
- How CLM solves the surface energy budget
- How CLM and ParFlow are coupled – what parts of the system are solved by ParFlow and what by CLM
- What is the difference between the version of CLM in Parflow and other CLM version
- What additional inputs are required to run ParFlow-CLM
- What additional outputs are generated when you run ParFlow-CLM
- How to ‘turn on’ CLM in ParFlow
- Additional input files needed for ParFlow-CLM
- Additional outputs generated by ParFlow-CLM

Where did CLM originate?

(Common Land Model)

- Community effort to combine best pieces of existing modular land surface models

- 3 models
  1. Land Surface Model (LSM) Bonan (1996)
  2. Biosphere Atmosphere Transfer Scheme (BATS) Dickinson (1993)
  3. Chinese Academy of Sciences Institute of Atmospheric Physics LSM 1994 version (IAP94) Dai and Zeng (1997)

- Initial documentation in Dai et al. (2003)

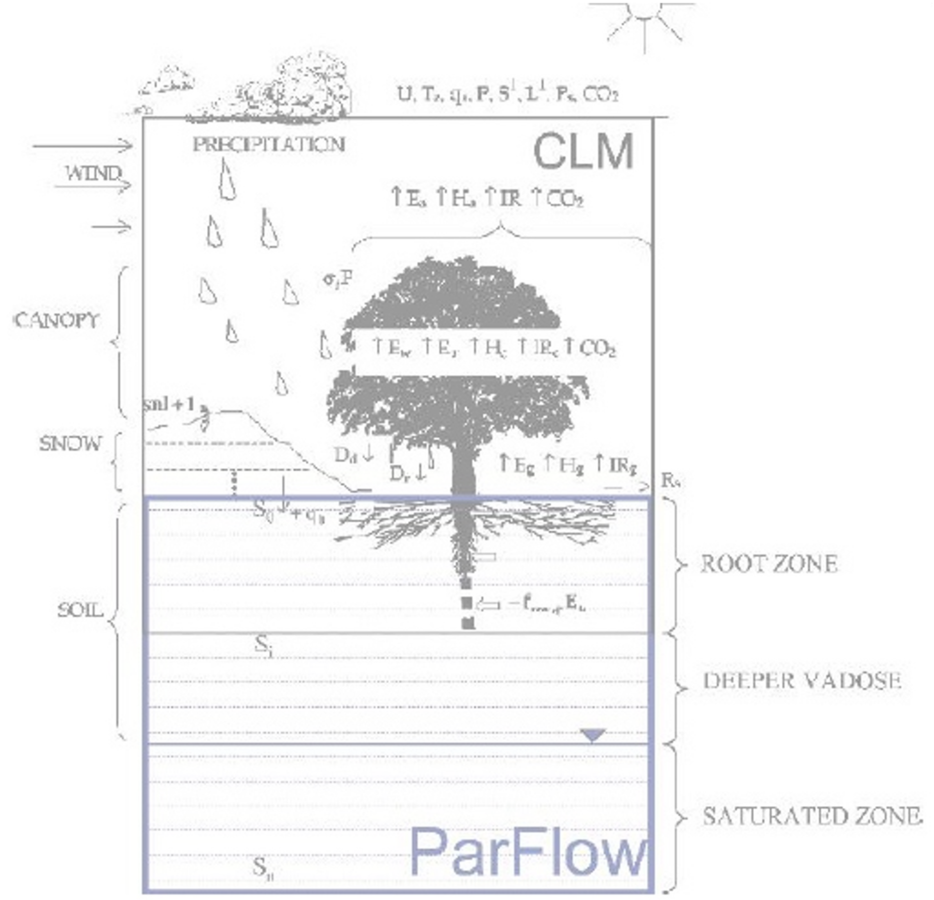
Who maintains CLM today?

- Name change from “Common” to “Community” occurred around 2002 with the release of CLMv2
- National Center for Atmospheric Research (NCAR)
- CLM is now the land surface component of the Community Earth System Model (CESM)

- *Why is this so confusing?*

When was CLM first coupled to PF?

- Maxwell and Miller (2005) – *Development of a Coupled Land Surface and Groundwater Model*
- Kollet and Maxwell (2008)— *Capturing the influence of groundwater dynamics on land surface processes using an integrated, distributed watershed model*
  - PF replaced CLM soil moisture formulation
  - Surface (CLM) formulations remained the same
- ≈CLMv3
- PF and CLM communicate over any number of soil layers
- Fluxes and variables passed between models at every time step



For more applications/papers see Table 1.1 in ParFlow Manual

What happens in CLM?

A lot of calculations!

➡ See the PF-CLM flowchart

http://parflow.blogspot.com/2015/10/clm-modules.html

I would like references on how CLM processes work, what are the best ones?

Common Land Model (CLM)
Surface

Subsurface
ParFlow (PF)

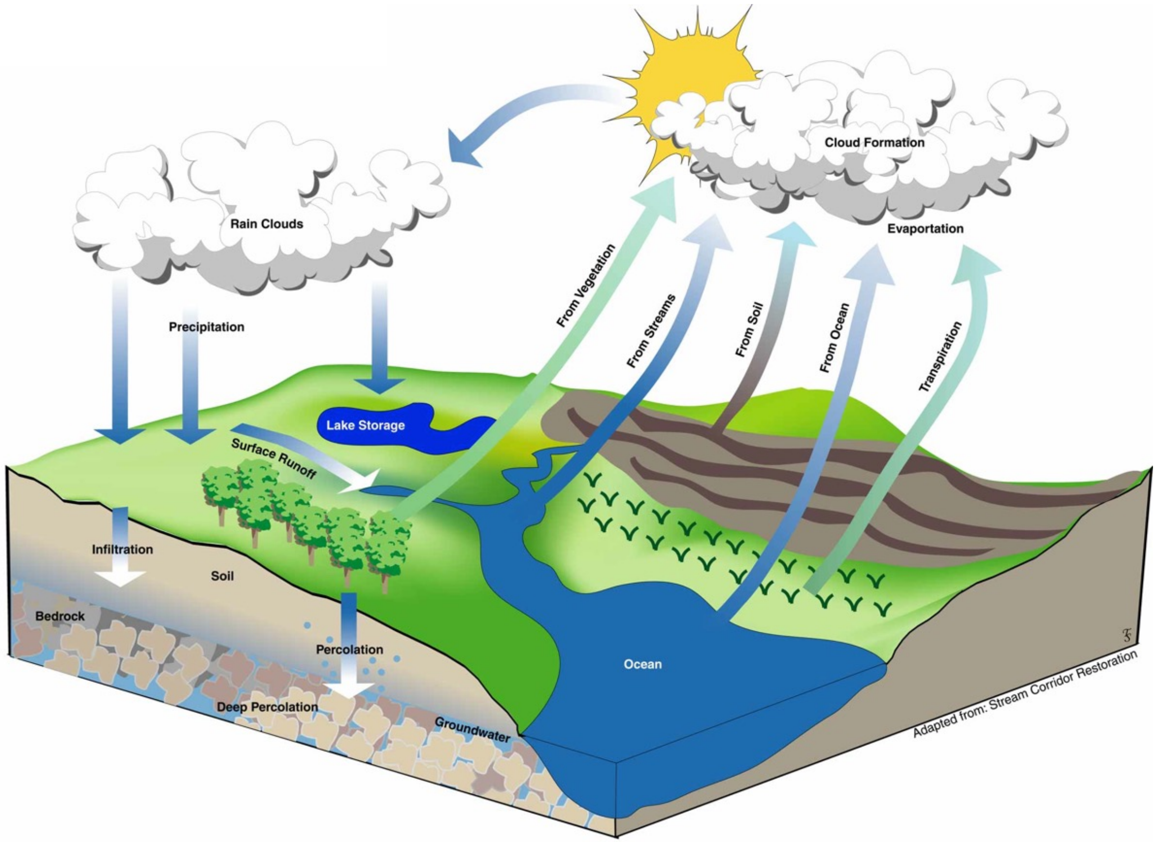
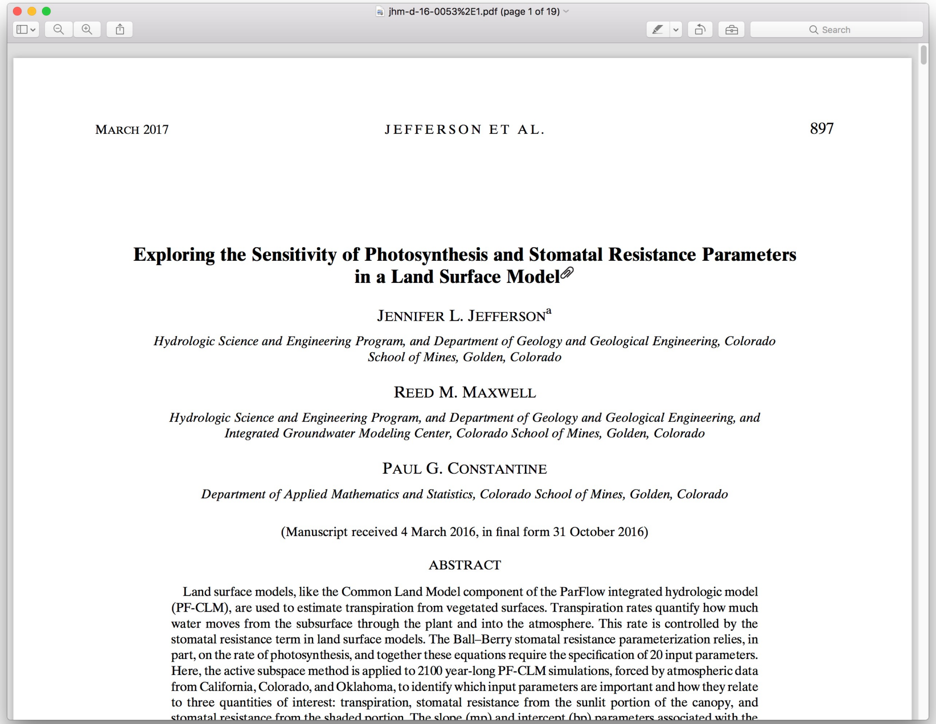


Image Source: http://www.buffer.forestry.iastate.edu/Photogallery/illustrations/illustrations-1.htm

Here are a few references on the CLM in ParFlow

- Kollet, et al. The Influence of Rain Sensible Heat and Subsurface Energy Transport on the Energy Balance at the Land Surface. *Vadose Zone Journal* 2009.
- Jefferson and Maxwell, Evaluation of simple to complex parameterizations of bare ground evaporation. *Journal of Advances in Modeling Earth Systems* 2015.
- Ferguson et al. Effects of Root Water Uptake Formulation on Simulated Water and Energy Budgets at Local and Basin Scales. *Environmental Earth Sciences* 2016.
- Jefferson et al. Exploring the Sensitivity of Photosynthesis and Stomatal Resistance Parameters in a Land Surface Model. *Journal of Hydrometeorology* 2017.
- Ryken et al. Sensitivity and model reduction of simulated snow processes: contrasting observational and parameter uncertainty to improve prediction *Advances in Water Resources* 2020.



MARCH 2017 JEFFERSON ET AL. 897

Exploring the Sensitivity of Photosynthesis and Stomatal Resistance Parameters in a Land Surface Model

JENNIFER L. JEFFERSON[a]

Hydrologic Science and Engineering Program, and Department of Geology and Geological Engineering, Colorado School of Mines, Golden, Colorado

REED M. MAXWELL

Hydrologic Science and Engineering Program, and Department of Geology and Geological Engineering, and Integrated Groundwater Modeling Center, Colorado School of Mines, Golden, Colorado

PAUL G. CONSTANTINE

Department of Applied Mathematics and Statistics, Colorado School of Mines, Golden, Colorado

(Manuscript received 4 March 2016, in final form 31 October 2016)

ABSTRACT

Land surface models, like the Common Land Model component of the ParFlow integrated hydrologic model (PF-CLM), are used to estimate transpiration from vegetated surfaces. Transpiration rates quantify how much water moves from the subsurface through the plant and into the atmosphere. This rate is controlled by the stomatal resistance term in land surface models. The Ball–Berry stomatal resistance parameterization relies, in part, on the rate of photosynthesis, and together these equations require the specification of 20 input parameters. Here, the active subspace method is applied to 2100 year-long PF-CLM simulations, forced by atmospheric data from California, Colorado, and Oklahoma, to identify which input parameters are important and how they relate to three quantities of interest: transpiration, stomatal resistance from the sunlit portion of the canopy, and stomatal resistance from the shaded portion. The slope (mp) and intercept (bp) parameters associated with the

Setting up a ParFlow-CLM model: Little Washita Watsherd Example

Additional setup for PF-CLM

- Additional files inputs the model will need
    1. drv_vegm.dat
    2. drv_vegp.dat
    3. drv_clmin.dat
    4. Meteorological forcing file(s) – 1D or 3D

- Before you start you will need to have IGBP land cover classifications determined for every grid cell in your domain

drv_vegm.dat

(includes information for each tile in domain)

x, y coordinate for each tile in domain; coordinates for single column (1, 1) are shown

cosine of the zenith angle (light for photosynthesis/transpiration)
convert from GMT to local time

soil thermal properties

soil albedo calculation; scale of 1 (light) to 8 (dark)

```
x   y   lat      lon         sand  clay     color
        (Deg)    (Deg)       (%/100)        index
1   1   38.4316 -120.9660   0.16  0.265    2

fractional coverage of grid by vegetation class (Must/Should Add to 1.0)
1    2    3    4    5    6    7    8    9    10   11   12   13   14   15   16   17   18
0.0 0.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
```

Remember, PF-CLM does not have fractional coverage!

Land Cover Types
(i.e., vegetation class in drv_vegp.dat)

(IGBP = International Geosphere-Biosphere Programme)

```
!IGBP Land Cover Types (other classes can be used by changing this file)
!  1 evergreen needleleaf forests
!  2 evergreen broadleaf forests
!  3 deciduous needleleaf forests
!  4 deciduous broadleaf forests
!  5 mixed forests
!  6 closed shrublands
!  7 open shrublands
!  8 woody savannas
!  9 svannas
! 10 grasslands
! 11 permanent wetlands
! 12 croplands
! 13 urban and built-up lands
! 14 cropland / natural vegetation mosaics
! 15 snow and ice
! 16 barren or sparsely vegetated
! 17 water bodies
! 18 bare soil
```

drv_vegp.dat
(specifies vegetation parameter values)

```
lai             Maximum leaf area index [-]
6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 2.00 6.00 6.00 5.00 6.00 0.00 6.00 0.00 0.00
!
lai0            Minimum leaf area index [-]
5.00 5.00 1.00 1.00 3.00 2.00 1.00 2.00 1.00 0.50 0.50 0.50 1.00 2.00 0.00 0.50 0.00 0.00
```

lai properties that correspond to 10. grasslands

- (maximum) leaf area index (-)
- (minimum) leaf area index (-)
- stem area index (-)
- aerodynamic roughness length (m)
- displacement height (m)
- leaf dimension (m)
- fitted numerical index of rooting distribution (-)
- fitted numerical index of rooting distribution (-)

- leaf reflectance – visible light (-)
- leaf reflectance – near infrared light (-)
- stem reflectance – visible light (-)
- stem reflectance – near infrared light (-)
- leaf transmittance – visible light (-)
- leaf transmittance – near infrared light (-)
- stem transmittance – visible light (-)
- stem transmittance – near infrared light (-)
- leaf/stem orientation index (-)
- btran exponent (-)

drv_clmin.dat

(includes timing information and additional parameters)

- Make sure times are entered in GMT and correspond to times in the forcing file
- Change name of output file prefixes, if desired

```
outf1d        stomataSA.output.txt             CLM output file
poutf1d       stomataSA.para.out.dat           CLM 1D Parameter Output File
rstf          stomataSA.rst.                   CLM active restart file
```

- Update lines in this file if restarting simulation (from 2 to 1)

```
startcode     2                                 1=restart file,2=defined
clm_ic        2                                 1=restart file,2=defined
```

- Includes several other specified parameter values
  - Meteorological station heights
  - Roughness lengths
  - Finite difference parameters

Meteorological Forcing File
(Pg. 140 ParFlow Manual)

DSWR: Downward Visible or Short-Wave radiation $[W/m^2]$.

DLWR: Downward Infa-Red or Long-Wave radiation $[W/m^2]$

APCP: Precipitation rate $[mm/s]$

Temp: Air temperature $[K]$

UGRD: West-to-East or U-component of wind $[m/s]$

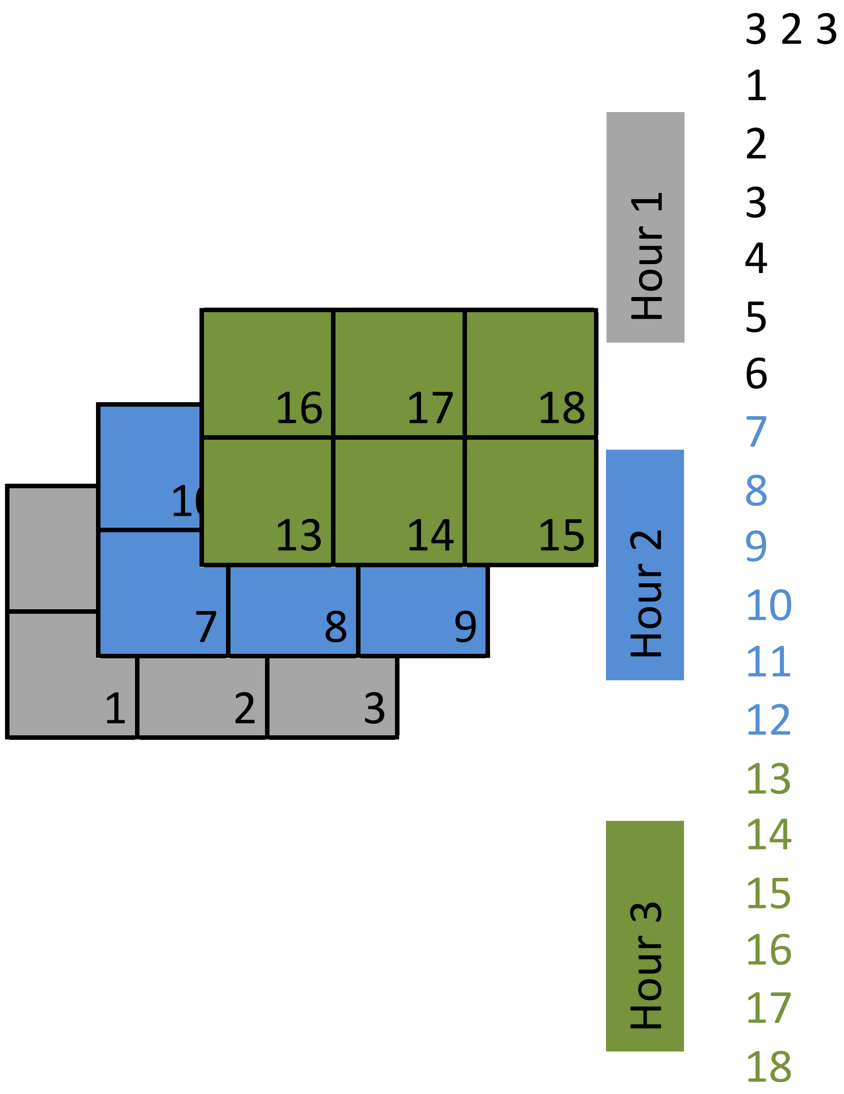
VGRD: South-to-North or V-component of wind $[m/s]$

Press: Atmospheric Pressure $[pa]$

SPFH: Water-vapor specific humidity $[kg/kg]$

- Columns must be in this order and have these units
- 1D – .txt file with single column for each variable and each row is one timestep
- 3D - .pfb files, one for each variable and multiple time steps

3D Forcing Files PF-CLM



- Separate files for every forcing variable
- You can put multiple hours in every forcing file. Time is the z dimension
- Tell CLM how many hours are in each forcing files using CLM.MetFileNT

Additional setup for PF-CLM

- Distribute your forcing files **before** you run and remember that the nz is the number of time steps per file for 3D forcing inputs

- See Dist_Forcings.tcl script in the Washita test case folder for an example

- Remember that if you change your processor topology you <u>must</u> redistribute your forcings

How do I "turn-on" CLM in PF?

```
model.Solver.LSM = "CLM"
```

- Optional CLM Flags are listed in the ParFlow Manual 6.1.36

Adding CLM settings to your tcl script:

```
#outputs
model.Solver.LSM = "CLM"
model.Solver.CLM.CLMFileDir = "clm_output/"
model.Solver.CLM.Print1dOut = False
model.Solver.BinaryOutDir = False
model.Solver.CLM.CLMDumpInterval = 1
Model.Solver.PrintCLM = True
model.Solver.CLM.DailyRST = True
model.Solver.CLM.SingleFile = True

#forcing files
model.Solver.CLM.MetForcing = "3D"
model.Solver.CLM.MetFileName = "NLDAS"
model.Solver.CLM.MetFilePath = "../../NLDAS"
model.Solver.CLM.MetFileNT = 24
model.Solver.CLM.IstepStart = 1

# physical properties
model.Solver.CLM.EvapBeta = "Linear"
model.Solver.CLM.VegWaterStress = "Saturation"
model.Solver.CLM.ResSat = 0.1
model.Solver.CLM.WiltingPoint = 0.12
model.Solver.CLM.FieldCapacity = 0.98
model.Solver.CLM.IrrigationType = "none"
model.Solver.CLM.RootZoneNZ = 4
model.Solver.CLM.SoiLayer = 4
```

What information is output from CLM?

(Order of information obtained from solver_richards.c)

single file output layer

```
1. eflx_lh_tot                ! latent heat flux from canopy height to atmosphere [W/2]
2. eflx_lwrad_out             ! outgoing long-wave radiation from ground+canopy
3. eflx_sh_tot                ! sensible heat from canopy height to atmosphere [W/m2]
4. eflx_soil_grnd             ! ground heat flux [W/m2]
5. qflx_evap_tot              ! evapotranspiration from canopy height to atmosphere [mm/s]
6. qflx_evap_grnd             ! ground surface evaporation rate (mm h2o/s)
7. qflx_evap_soi              ! evaporation heat flux from ground [mm/s]
8. qflx_evap_veg              ! evaporation+transpiration from leaves [mm/s]
9. qflx_tran_veg              ! transpiration rate [mm/s]
10. qflx_infl                 ! infiltration (mm H2O /s)
11. swe_out                   ! snow water equivalent
12. t_grnd                    ! ground temperature (K)

13. qflx_qirr                 ! qflx_surf directed to irrig (mm H2O/s);irrigation applied at
                          surface [mm/s](added to rain or throughfall, depending)
OR
13. qflx_qirr_inst(nlevsoi)   !irrigation applied by 'instant' method [mm/s] (added to pf_flux)

14-23. tsoil                  ! soil temperature for each soil layer; assuming 10 soil layers
```


ParFlow Short Course: ParFlow-CLM

How is this information output from CLM?

- Binary format
  - Can view .pfb files in paraview
  - use pf.read_pfb() to read .pfb files into python

- Single file output means 1 file for each time step that contains *all* variables on previous slide

  model.Solver.CLM.SingleFile **= True**

  Output files would be titled as follows:

  ```
  runnamethatyoupick.out.clm_output.00001.C.pfb
  runnamethatyoupick.out.clm_output.00002.C.pfb
  ...
  ```

- Non-single file output means 1 file for each variable for each time step

Common Pitfalls

- Remember that CLM will assume you have 10 soil layers. If this is not true refer to the CLM notes on how to adjust this
- All timing info in CLM should be in <u>GMT</u> not local time
- If you want to run with timesteps that don't match the forcing timestep you need to use the Solver.CLM.ReuseCount key. Changing the timestep in ParFlow does not automatically change the forcings

Starting a New Run

1. Set your initial pressure from the final pressure from your spinup

```
model.ICPressure.Type = "PFBFile"
model.ICPressure.GeomNames = "domain"
model.Geom.domain.ICPressure.RefPatch = "top"
model.Geom.domain.ICPressure.FileName = "press.init.pfb"
```

2. Setup your timing in ParFlow

```
model.TimingInfo.BaseUnit = 1.0
model.TimingInfo.StartCount = 0
model.TimingInfo.StartTime = 0
model.TimingInfo.StopTime = 72.0
model.TimingInfo.DumpInterval = 1.0
model.TimeStep.Type = "Constant"
model.TimeStep.Value = 1.0

Model.Solver.CLM.IstepStart = 1.0
```

Note ParFlow starts at 0, which is the initial condition, and the CLM.IstepStart is 1 which is the first point an output will be written for

Starting a New Run

3. Tell CLM to start from scratch and define the starting time in drv_clmin.dat

```
startcode       2                         1=restart file,2=defined
clm_ic          2                         1=restart file,2=defined

sss             00                        Starting Second
smn             00                        Starting Minute
shr             00                        Starting Hour
sda             01                        Starting Day
smo             10                        Starting Month
syr             1998                      Starting Year
```

** Remember that all times in the drv_clmin.dat file are in GMT

Starting a New Run

4. Set your processor topology and make sure you are distributing all of your input files

```
model.Process.Topology.P = 1
model.Process.Topology.Q = 1
model.Process.Topology.R = 1

model.ComputationalGrid.NX = 64
model.ComputationalGrid.NY = 32
model.ComputationalGrid.NZ = 1
model.dist("slopex_LW.pfb")
model.dist("slopey_LW.pfb")

model.ComputationalGrid.NZ = 10
model.dist("Indicator_LW_USGS_Bedrock.pfb")
model.dist("press.init.pfb")
```

- Every pfb input file must be distributed
- Remember that slope files are 2D so NZ must be set to 1 before distributing
- Don’t forget to distribute your forcings separately
- You can’t redistribute clm restart files in the middle of a run so the topology you pick you should stick with

ParFlow Short Course: ParFlow-CLM
ParFlow Short Course: ParFlow-CLM

Restarting

1. Determine the last timestep that a CLM restart file was written for this is where you should restart from
2. Update the timing in ParFlow to reflect your new start point which should be equal to the time of the last restart file

```
model.TimingInfo.StartCount = 19.0
model.TimingInfo.StartTime = 19.0
model.TimingInfo.StopTime = 72.0
Model.Solver.CLM.IstepStart = 20.0
```

Again note that the CLM counter should start at 1 + startcount

Restarting

3. Overwrite your initial pressure file with the last pressure file output

```
cp pfclm.out.press.00018.pfb press.in.pfb
```

4. Change the restart settings in drv_clmin.dat

```
startcode        1                        1=restart file,2=defined
clm_ic           1                        1=restart file,2=defined
```

Note: when you restart a simulation (i.e., startcode = 1 and clm_ic = 1 in drv_clmin.dat), the timing information is read from the restart file not from drv_clmin.dat. However, if startcode = 2 and clm_ic = 1 the timing information will be read from drv_clmin.dat and initial condition information will still come from the restart file.

Restarting

5. If you have set CLM to overwrite the restart files as it goes (i.e. if CLM.WriteLastRST = TRUE), then copy restart files before you start

```
model.Solver.CLM.WriteLastRS = True

#CLM RESTART INFO
if { $startcount > 1 } then {

 for { set i 0 } { $i < $nproc } { incr i 1 } {
        set fname_rst [format "clm.rst.%05d.$i" [expr $startcount]]
        exec cp clm.rst.00000.$i $fname_rst
 }
}
```

Extra slides

What are *some* differences between PF-CLM and CLMv4.5?

- Soil resistance
  - PF-CLM incorporates soil moisture computed using 3D Richards equation
  - Choose between linear and cosine soil resistance factors to limit bare soil evaporation in PF-CLM
  - CLMv4.5 has vertical soil moisture transport (i.e, no lateral flow)

- Fractional vegetation coverage
  - PF-CLM is not setup to handle fractional vegetation (even if you put it in drv_vegm.dat this way)
  - CLMv4.5 tiles can have several land uses

What are *some* differences between PF-CLM and CLMv4.5?

- Leaf area index (LAI)
  - PF-CLM computes LAI at each time step using an empirical equation that depends on soil temperature
  - CLMv4.5 updates LAI daily based on interpolation of monthly MODIS LAI values
- ET adjustment factors
  - PF-CLM assumes C3 plants (unless manually changed)
  - CLMv4.5 includes additional factors to adjust photosynthesis rates/stomatal resistance (canopy scaling, nitrogen, day length)
- Irrigation through Water Allocation Model (WAM)

Where does PF call CLM?

from solver_richards.c

```
CALL_CLM_LSM(pp,sp,et,ms,po_dat,dz_dat,istep,cdt,t,start_time,
                            dx,dy,dz,ix,iy,nx,ny,nz,nx_f,ny_f,nz_f,nz_rz,ip,p,q,r,gnx, gny,rank,
                            sw_data,lw_data,prcp_data,tas_data,u_data,v_data,patm_data,qatm_data,
                               eflx_lh,eflx_lwrad,eflx_sh,eflx_grnd,qflx_tot,qflx_grnd,
                            qflx_soi,qflx_eveg,qflx_tveg,qflx_in,swe,t_g,t_soi,
                               public_xtra -> clm_dump_interval,
                               public_xtra -> clm_1d_out,
                               public_xtra -> clm_file_dir,
                               clm_file_dir_length,
                               public_xtra -> clm_bin_out_dir,
                               public_xtra -> write_CLM_binary,
                               public_xtra -> clm_beta_function,
                               public_xtra -> clm_veg_function,
                               public_xtra -> clm_veg_wilting,
                               public_xtra -> clm_veg_fieldc,
                               public_xtra -> clm_res_sat,
                               public_xtra -> clm_irr_type,
                               public_xtra -> clm_irr_cycle,
                               public_xtra -> clm_irr_rate,
                               public_xtra -> clm_irr_start,
                               public_xtra -> clm_irr_stop,
                               public_xtra -> clm_irr_threshold,
                               qirr, qirr_inst, iflag,
                               public_xtra -> clm_irr_thresholdtype,
                               soi_z,clm_next,clm_write_logs,clm_last_rst,clm_daily_rst);
```

This seems like more detail than is needed for an intro course we don't show where anything else lives in the source code I vote delete

all of these variables get passed from PF to CLM

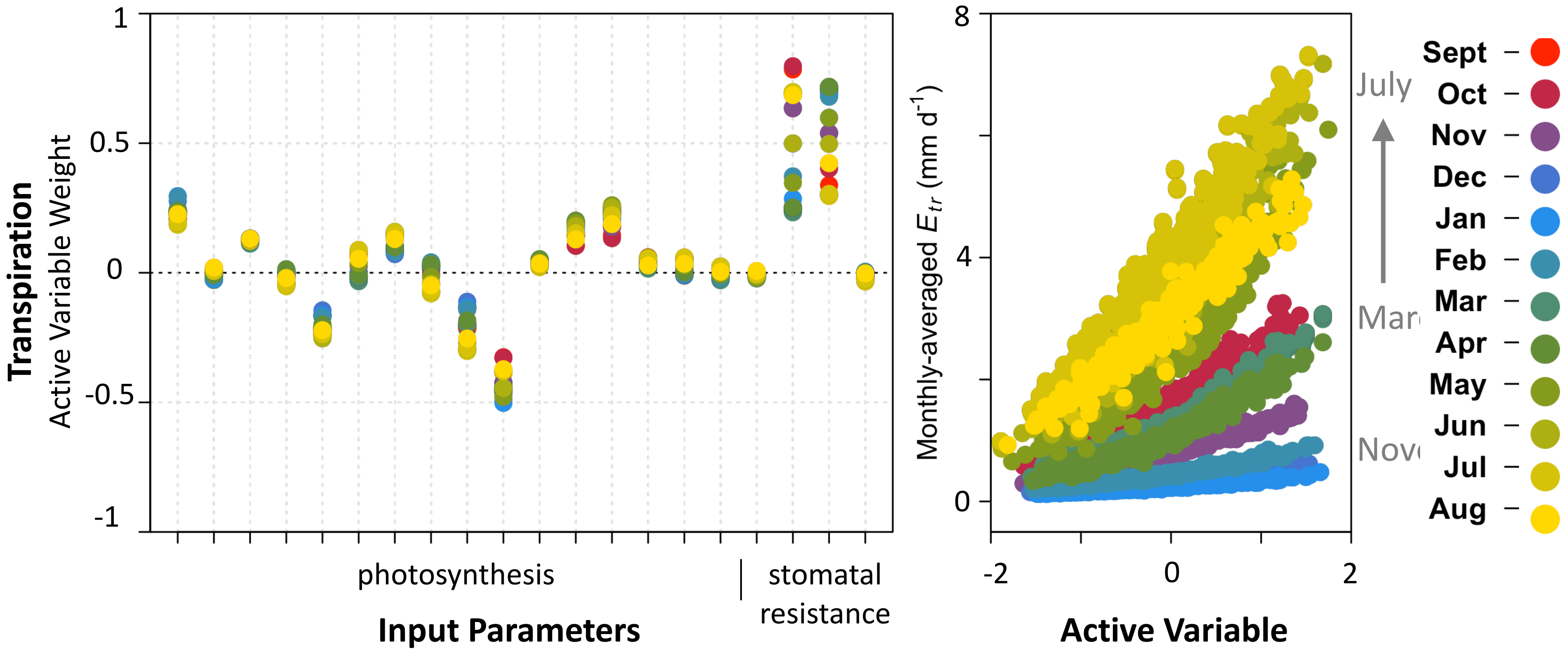
Input parameters used to compute stomatal resistance

photo-synthesis

stomatal resistance

Parameter Description	Name	Distribution (Range)	Default value	Units
maximum rate of carboxylation at 25°C	**vcmx25**	U(20, 65)	33	μmol CO_2 m^{-2} s^{-1}
q10 for vcmx25	avcmx	U(2.2, 2.6)	2.4	-
deactivation energy constant	hv	U(218,000, 242,000)	220,000	J mol^{-1}
entropy constant	sv	U(640, 730)	710	J mol^{-1} K^{-1}
CO_2 Michaelis-Menten constant at 25°C	**kc25**	U(25, 50)	30	Pa
q10 for kc25	akc	U(1.9, 2.3)	2.1	-
O_2 Michaelis-Menten constant at 25°C	ko25	U(30,000, 45,000)	30,000	Pa
q10 for ko25	ako	U(1.1, 1.3)	1.2	-
maximum ratio of oxygenation to carboxylation	**ocr**	U(0.18, 0.77)	0.21	-
c_i mulitplier in denominator of wj	**wj1**	1, 4, 4.5	1	-
cp mulitplier in denominator of wj	wj2	2, 8, 10.5	2	-
energy content of photons	ecp	U(3.3, 5.8)	4.6	μmol J^{-1}
quantum efficiency at 25°C	qe25	U(0.04, 0.08)	0.06	μmol CO_2 μmol $photon^{-1}$
multiplier in we	we1	U(0.45, 0.55)	0.5	-
partial pressure of CO_2 in the atmosphere	ppcd	U(355, 400)	355	ppm
ratio of diffusivity of CO_2 to H_2O in boundary layer	drb	U(1.3, 1.4)	1.37	-
ratio of diffusivity of CO_2 to H_2O through stomata	drs	U(1.6, 1.7)	1.65	-
minimum leaf conductance	**bp**	U(1,000, 10,000)	2,000	μmol m^{-2} s^{-1}
slope for conductance-to-photosynthesis relationship	**mp**	U(4, 12)	9	-
maximum stomatal resistance	rsmax0	U(10,000, 40,000)	20,000	s m^{-1}

Weights and sufficient summary plots for monthly-averaged transpiration vary seasonally



Source: Jefferson et al. (in preparation)

Variables in CLM

- Global variables
  - clm%zlnd
  - See clmtype.F90
- Local variables
  - efpot
  - See individual modules
- Constant values
  - Gravity = 9.8616
  - See clm_varcon.F90

This section also seems like more detail than is needed for intro. I would recommend shortening or deleting