Setting up a watershed model: Little Washita Example part 2

ParFlow Short Course

Workflow Outline

1. Evaluate available model inputs
2. Determine your domain configuration
3. Process topography
4. Setup the subsurface
5. Initialize the model (i.e. spinup)
6. Additional setup for PF-CLM

This is also outlined in section 3.1.2 of the manual

Subsurface inputs

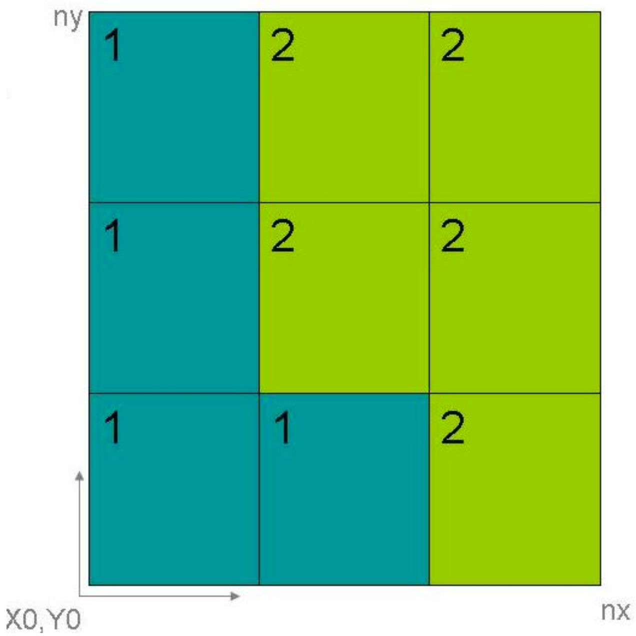
1. Identify unique subsurface units and provide subsurface geometries to ParFlow
   - Either using solid files or indicator files

2. Assign the hydrologic properties to each unit
   - permeability, specific storage, porosity and van Genuchten parameters

http://parflow.blogspot.com/2015/08/subsurface-setup-options.html

Indicator files

- 3D ParFlow file that has a integer assignment for every grid cell designating what subsurface unit the cell belongs to



http://parflow.blogspot.com/2007/09/using-indicator-files-as-input-to.html

Subsurface inputs: Indicator files

1. Tell Parflow that you will be using and indicator file and give unit a name

```
pfset GeomInput.indi_input.InputType      IndicatorField
pfset GeomInput.indi_input.GeomNames      "s1 s2 s3 g1 g2 g3”
pfset Geom.indi_input.FileName            "IndicatorFile..pfb"
```

2. Match every unit name with an integer value in the indicator file

```
pfset GeomInput.s1.Value                1
pfset GeomInput.s2.Value                2
pfset GeomInput.s3.Value                3
pfset GeomInput.g1.Value                21
pfset GeomInput.g2.Value                22
pfset GeomInput.g3.Value                23
```

http://parflow.blogspot.com/2015/08/subsurface-setup-options.html

Assigning Properties

- Provide a list of the geometries you will be assigning values to and for <u>every</u> geometry listed provide a value

- You can set a background value using your domain geometry and you don’t have to use all of the indicator geometries you defined as long as every cell has a value

```
pfset Geom.Perm.Names                  "domain s1 s2"

pfset Geom.domain.Perm.Type             Constant
pfset Geom.domain.Perm.Value            0.2

pfset Geom.s1.Perm.Type                  Constant
pfset Geom.s1.Perm.Value                 0.269022595

pfset Geom.s2.Perm.Type                  Constant
pfset Geom.s2.Perm.Value                 0.043630356
```

Assigning Properties

- Repeat the process for the other variables
- You can use different geometry combinations for different variables

```
pfset Geom.Porosity.GeomNames                "domain s1 s2 s3  g1 g2 g3"
pfset Geom.domain.Porosity.Type             Constant
pfset Geom.domain.Porosity.Value            0.4
...

pfset Phase.RelPerm.Type                     VanGenuchten
pfset Phase.RelPerm.GeomNames                "domain"
pfset Geom.domain.RelPerm.Alpha              3.5
pfset Geom.domain.RelPerm.N                  2.0
...

pfset Phase.Saturation.Type                  VanGenuchten
pfset Phase.Saturation.GeomNames             "domain s1 s2 s3 "
pfset Geom.domain.Saturation.Alpha           3.5
pfset Geom.domain.Saturation.N               2.
pfset Geom.domain.Saturation.SRes            0.2
pfset Geom.domain.Saturation.SSat            1.0
```

Adding Heterogeneity

- Turning bands method is built into ParFlow
- To implement you will need some preliminary analysis on the statistical properties of your domain
- You can implement turning bands separately for different subsurface units within the domain
- Refer to the Harvey flow example section 3.6.1 in the manual

```
pfset Geom.domain.Perm.Type                  "TurnBands”
pfset Geom. domain.Perm.LambdaX              3.60
pfset Geom. domain.Perm.LambdaY              3.60
pfset Geom. domainPerm.LambdaZ               0.19
pfset Geom. domain.Perm.GeomMean             112.00
pfset Geom.upper_aquifer.Perm.Sigma             0.48989794
pfset Geom.upper_aquifer.Perm.NumLines             150
pfset Geom.upper_aquifer.Perm.Seed                 33333
pfset Geom.upper_aquifer.Perm.LogNormal            Log
```

5. Initializing the model (spinup)

Determining the starting groundwater configuration for your simulations

Spinup

- There is no best practice for spinup, results and approaches will vary depending on your domain and the questions you want to answer with your model
- The goal is to have a domain that is stable (metrics for this may also vary) and solving nicely before you start making runs to answer questions
- Groundwater is the slowest moving part so its often easiest to start with a simplified system and get a stable water table before adding in land surface processes

http://parflow.blogspot.com/2015/08/spinning-up-watershed-model.html

One Approach to Spinup

1. Initialize your water table somewhere

```
You can set the water table to a constant depth like this:
pfset
ICPressure.Type                                   HydroStaticPatch
pfset ICPressure.GeomNames                        domain
pfset Geom.domain.ICPressure.Value                -10.0
pfset Geom.domain.ICPressure.RefPatch             z-upper


or start your domain off completely dry like this:
pfset
ICPressure.Type                                   HydroStaticPatch
pfset ICPressure.GeomNames                        domain
pfset Geom.domain.ICPressure.Value                0.0
pfset Geom.domain.ICPressure.RefPatch             z-lower


or you can read in an input file with pressure heads like this:
pfset ICPressure.Type                             PFBFile
pfset ICPressure.GeomNames                        domain
pfset Geom.domain.ICPressure.FileName              press.in.pfb
```

One Approach to Spinup

2. Run for a long time with a constant recharge forcing and no surface water flow

- You can set a spatially variable flux using a pfb file like this or you can set a homogenous flux using the options shown for the parking lot test

```
pfset Solver.EvapTransFile                True
pfset Solver.EvapTrans.FileName           “PmE.flux.pfb“
pfdist                                    "PmE.flux.pfb"
```

- The units of this flux should be 1/T so if you have a flux that is L/T remember to divide by the thickness of each layer
- This is a 3D file so if you just want a flux applied to the top of the domain set the values for all other layers to zero

Spinup: One approach

2. Run for a long time with a constant recharge forcing and no surface water flow

- You can turn off overland flow like this

```
pfset OverlandFlowSpinUp              1
```

- This key removes any ponded surface water at every time step so that no overland flow occurs.
- This is just a trick to make the problem easier to solve in the beginning and should <u>not</u> be used for regular simulation.

Spinup: One Approach

3. Turn overland flow back on and run until streams have formed and overland flow is stabilized

```
pfset OverlandFlowSpinUp            0
```

How do I know if I’m done?

Spinup Analysis

Look at your outputs and check if you are converging.

- Is your water table changing between time steps?
- Is groundwater storage changing?
- Are your surface water outflows changing?

Spinup Analysis

Look at your outputs and check if you are converging.

- You can do this visually with Visit

- You should also calculate your water balance and outflows at important locations in your domain

  (see examples 6-9 in section 4.3 of the ParFlow Manual for how to do this with PFTools)

What do I do if I’m stuck?

Getting stuck in spinup is very common. It can be challenging and will likely require close monitoring and frequent interventions. If you are stuck some of the first things to do are:

1. Look at your outputs and make sure you don’t have something wrong with your domain.

2. Look at the kinsol.log file and see how the model is solving

The Kinsol Log File

This is a very important file! It keeps track of the ParFlow's progress as it converges to a solution and can tell you a lot about how hard it is working and when its getting stuck.

NOTE: If you don't have output files yet the model could still be working to solve the first time step. If you don't have a kinsol log then its not doing anything and you have a problem with your model installation or your inputs

http://parflow.blogspot.com/2015/08/kinsol-logs-and-troubleshooting-slow.html

The Kinsol Log File

```
KINSOL starting step for time 1.000000
scsteptol used:       1e-30
fnormtol  used:       1e-06
KINSolInit nni=    0  fnorm=        2859.631546316827  nfe=     1
KINSol nni=    1 fnorm=        2798.943588424358 nfe=     5
        KINSol nni=    2 fnorm=        1264.208176165057 nfe=     8
KINSol nni=    3 fnorm=        1221.629858300943 nfe=    13
KINSol nni=    4 fnorm=        903.7712374454168 nfe=    15
KINSol nni=    5 fnorm=         408.050165635947 nfe=    18
KINSol nni=    6 fnorm=        363.6154118376153 nfe=    26
KINSol nni=    7 fnorm=        4.635468210055366 nfe=    27
KINSol nni=    8 fnorm=      0.06076322632612002 nfe=    28
KINSol nni=    9 fnorm=     5.862763215189762e-05 nfe=    29
KINSol nni=   10 fnorm=     7.663997383858169e-07 nfe=    30
KINSol return value 1
---KINSOL_SUCCESS


-------------------------------------------------
                    Iteration         Total
Nonlin. Its.:            10             10
Lin. Its.:              121            121
Func. Evals.:            30             30
PC Evals.:               10             10
PC Solves:              131            131
Lin. Conv. Fails:         0              0
Beta Cond. Fails:         0              0
Backtracks:               0              0
-------------------------------------------------
```

This tracks the model residual (fnorm) as the model converges.

Once fnorm falls below the tolerance set in your script *Solver.Nonlinear.ResidualTol* the time step is solved.

If…

1. The maximum number of iterations is reached *Solver.Nonlinear.MaxIter*
2. fnorm stops changing between iterations *Solver.Nonlinear.StepTol*

The time step is failed and ParFlow will half the time step and try again

The Kinsol Log File

- It’s normal for the model to require a lot of iterations to converge when its just getting started

- You should see the number of nonlinear iterations (Nonlin. Its.), linear iterations (Lin. Its.) and function evaluations (Func. Evals.) per time step decreasing as the model gets going.

Things to look for if it’s not solving

1. Areas of very high or very low pressure in your outputs
   - These could be physical or they could be a problem with your inputs. Either way they make the problem hard to solve.

2. Lots of places where overland flow is just staring to form.
   - Switching on and off overland flow creates a lot of work for the solver. If streams are just staring this can explain slow behavior.

Spinup Knobs to Turn

1. Use OverlandSpinup keys to dampen the pressure relationship in the overland flow equation with the following dampening term that gets added to the equation when pressures are less than zero:

   P2*exp(pressure*p1)

```
pfset OverlandSpinupDampP1        10.0
pfset OverlandSpinupDampP2        0.1
```

Spinup Knobs to Turn

2. Make your time step smaller or switch to a growth time step

```
pfset TimeStep.Type              Growth
pfset TimeStep.InitialStep       0.01
pfset TimeStep.GrowthFactor      1.1
pfset TimeStep.MaxStep           10000000
pfset TimeStep.MinStep           0.01
```

3. Experiment with different initial conditions
   - Sometimes it may solve better starting with the domain completely dry or completely wet rather than making a more intelligent initial guess

Spinup Knobs to Turn

4. Change your solver settings

Refer to the *manual section 6.1.33* for details and test scripts for examples. Some common options are:

- Change the preconditioner *(Solver.Linear.Preconditioner.SymmetricMat)*
- Increase the number of nonlinear iterations you allow (*Solver.Nonlinear.MaxIter*)
- Increase the convergence tolerance (*Solver.Nonlinear.ResidualTol*)
- NOTE: Increasing the number of processors will improve your run time but it does not fix problems that don't converge