ParFlow-PEST coupling

Reed Maxwell, rmaxwell@mines.edu

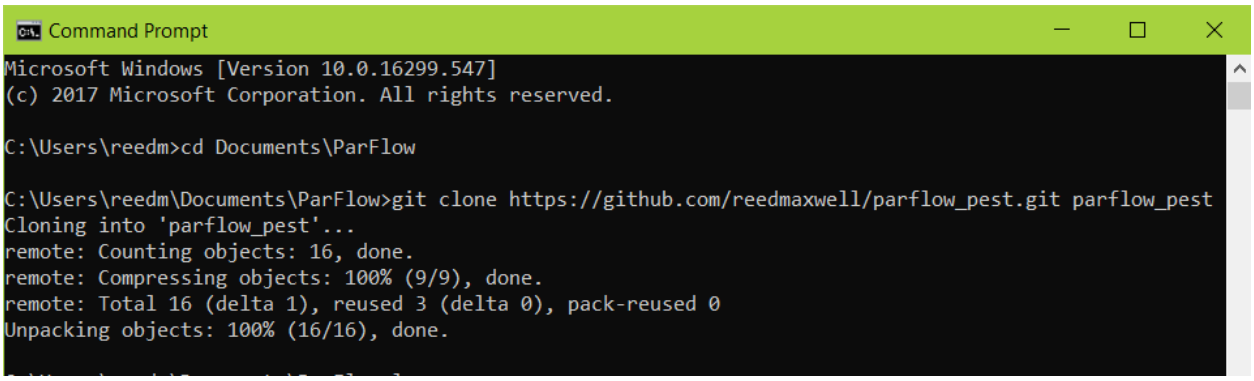
23-July-2018

The integrated hydrology model ParFlow has a number of features that make it easy to connect to other programs. The TCL/TK front end allows a lot of flexibility to read files and customize output. Here I describe how to couple ParFlow to PEST, one of the most popular parameter estimation programs out there. This works on Mac, PC or Linux; with ParFlow compiled locally or running in a Docker container.

1. Download the example from GitHub https://github.com/reedmaxwell/parflow_pest

This can be done either by grabbing a zip file or by checking it out from GitHub using git. I like checking it out from GitHub, I opened a command prompt, changed to my ParFlow scratch directory and typed

```
C:\Users\reedm\Documents\ParFlow>git clone
https://github.com/reedmaxwell/parflow_pest.git parflow_pest
```

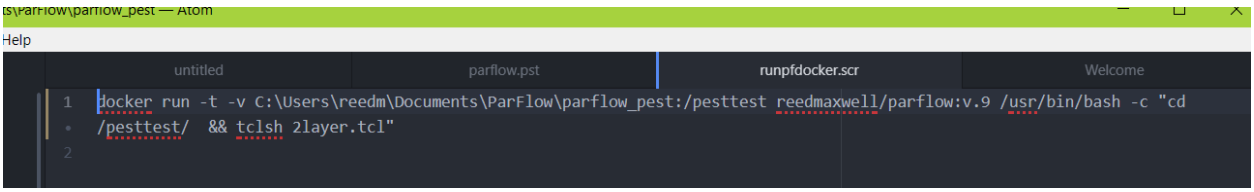


2. Download or install PEST. You can grab either the PEST executable or the source code from http://www.pesthomepage.org/Downloads.php

On Windows this downloads an executable in a zip file, on the Mac or Linux this downloads source code that builds relatively smoothly.

3. Edit the `parflow.pst` file and the run script for ParFlow.

This script either runs ParFlow locally or via the Docker container. On the Mac I ran ParFlow using the `runpf.scr` script, on Windows I used the `runpfdocker.bat` script which runs ParFlow via a Docker container ported to the current directory. Often you want to use a text editor to do this, I like Atom. https://atom.io/ but many different editors will work. We need to edit two files `parflow.pst` to set the model command for PEST and `runpfdocker.bat` to set the correct paths. In Atom I'll edit the batch script file first.



Notice that I've used the Docker -v option to map my local directory `C:\Users\reedm\Documents\ParFlow\parflow_pest` to a dummy directory `/pesttest`

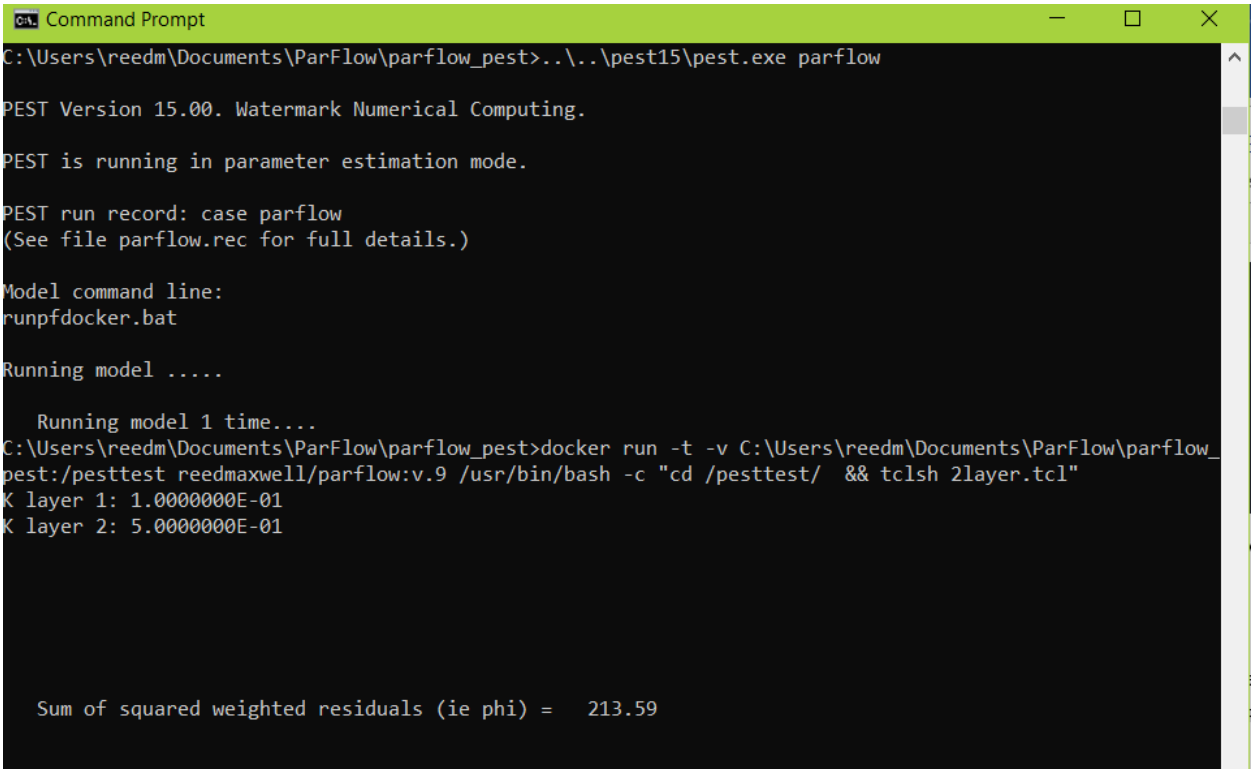
Your directories will be slightly different and you need to modify this file accordingly. If you want to test the ParFlow Docker install you can run the `runpfdocker.bat` right from the command line. If you are running on the Mac, you should be able to use the `runpf.scr` script directly. Next you want to edit the `parflow.pst` control file. Here you want to edit the *model section of the PEST control file to run the correct model.

```
21  p3            3.313235e+01   1.0  obsgroup
22  p4            2.872059e+01   1.0  obsgroup
23  p5            3.637714e+01   1.0  obsgroup
24  * model command line
25  runpfdocker.bat
26  * model input/output
27  in.tpl    input.txt
28  obs.pif   Press.txt
29
```

You can see that I swapped the run scripts, to run on Mac (or Linux) I'd keep this as `runpf.scr` but for Windows I'm swapping to `runpfdocker.bat`

4. Run PEST and ParFlow

Next we run PEST and ParFlow from the command line, I run from the current directory and point to where I have PEST installed. If you have PEST in your path you won't need this path.



The ParFlow model is a simple, synthetic, two-layer saturated simulation and PEST is estimating the hydraulic conductivity of each layer. The observation pressures are model-generated using the `true_K.txt` input. If you want to run the base case you can copy this file over to `input.txt`. The observations are written into the PEST control file. PEST writes it's calibrated hydraulic conductivity values into the `input.txt` which ParFlow reads in and assigns in the `2layer.tcl` script.

```
#-----------------------------------------------------------------------------
# Perm
#-----------------------------------------------------------------------------
pfset Geom.Perm.Names "upper_aquifer lower_aquifer"
# we open a file, in this case from PEST to set upper and lower aquifer values
# k1=upper_aquifer k2=lower_aquifer

set fileId [open input.txt r 0600]
set k1 [gets $fileId]
set k2 [gets $fileId]
close $fileId
## output K values used this run
puts "K layer 1: $k1"
puts "K layer 2: $k2"

## we use the parallel turning bands formulation in ParFlow to simulate
## GRF for upper and lower aquifer
##

pfset Geom.upper_aquifer.Perm.Type Constant
pfset Geom.lower_aquifer.Perm.Type Constant

#pfset lower aqu and upper aq stats to pest/read in values

pfset Geom.upper_aquifer.Perm.Value  $k1

pfset Geom.lower_aquifer.Perm.Value  $k2


pfset Perm.TensorType               TensorByGeom
```

ParFlow runs the simulation case with these values and then grabs the pressure head values output from the 2layer.out.press.pfb file, writing these to Press.txt.

```
# we use pf tools to load presssure
#
set press [pfload 2layer.out.press.pfb]

## uncomment if you want an ascii file of PF output every timestep
##pfsave $press -sa press.sa

# we use pftools to grab observation locations
# and write them to a file

set obs1 [pfgetelt $press 20 20 20]
set obs2 [pfgetelt $press 10 10 10]
set obs3 [pfgetelt $press 5 10 40]
set obs4 [pfgetelt $press 20 10 40]
set obs5 [pfgetelt $press 45 10 10]

set outfile [open Press.txt w]
# just the bare bones output to make things easier for PEST
#puts $outfile "I J K Pressure"
# 20 20 20
puts $outfile [format "%e" $obs1]
# 10 10 10
puts $outfile [format "%e" $obs2]
# 5 10 40
puts $outfile [format "%e" $obs3]
# 20 10 40
puts $outfile [format "%e" $obs4]
# 45 10 10
puts $outfile [format "%e" $obs5]

close $outfile
```

This allows you to modify this, or easily adapt this script to work with your problem.

```
        Phi =   44.131      (  0.209 of starting phi)

   No more lambdas: phi is less than 0.3000 of starting phi
   Lowest phi this iteration:   44.131
   Maximum   factor change:  3.000     ["k1"]
   Maximum relative change:  2.000     ["k1"]

   Optimisation complete:   3 optimisation iterations have elapsed since lowest
                         phi was achieved.
   Total model calls:     52

Running model one last time with best parameters.....
C:\Users\reedm\Documents\ParFlow\parflow_pest>docker run -t -v C:\Users\reedm\Documents\ParFlow\parfl
pest:/pesttest reedmaxwell/parflow:v.9 /usr/bin/bash -c "cd /pesttest/  && tclsh 2layer.tcl"
K layer 1: 1.0000000E-03
K layer 2: 7.7397953E-02









Recording run statistics .....

See file parflow.rec for full run details.
See file parflow.sen for parameter sensitivities.
See file parflow.seo for observation sensitivities.
See file parflow.res for residuals.

C:\Users\reedm\Documents\ParFlow\parflow_pest>dir
```