The Adjoint State Method: Highly Scalable Sensitivity Analysis

Jeremy White
Intera

Contributors and coauthors

Mohamed Hayek

Katie Markovich

Joe Hughes

Outline

1. Brief theory and overview of the adjoint method
2. Implementation details
3. Some examples

Why sensitivity analysis (again)?!

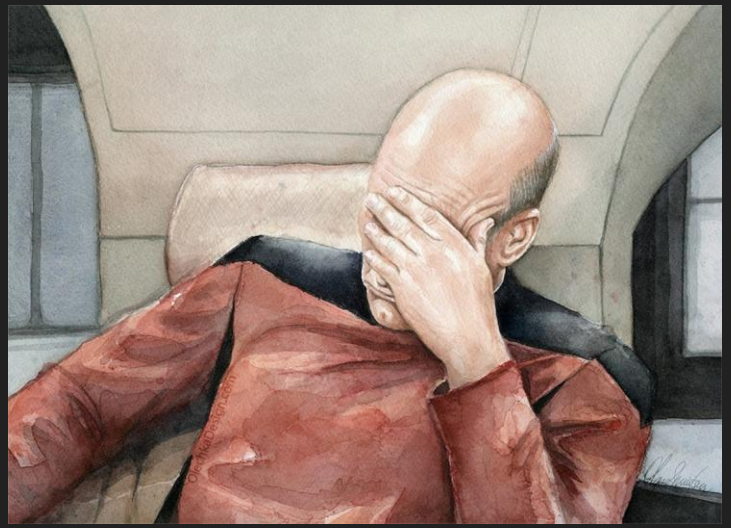
- Sensitivity analysis is a fundamental aspect of modeling building
- Develop “hydrosense”
- “What inputs are effecting these outputs?”
- “If I change model input X, what does that do?”
- Why not sensitivity analysis?
    - Its computationally expensive…

How?

- Run model at current parameter values
- For each parameter:
  - Change parameter some
  - Run model
  - Process results

How?

- Run model at current parameter values
- For each parameter:
  - Change parameter some
  - Run model
  - Process results



What is an adjoint solver

- An extreme efficient analytical gradient/sensitivity tool
- It’s been done before…
- Problem: its intrusive!
  - Traditionally, someone has to modify the forward model code
  - This leads to long-term maintenance and upkeep issues

Adjoint terminology

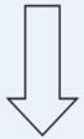
- “Performance measure”: a single or aggregate summary of model outputs of interest
  - Sum of weighted residuals (ie objective function)
  - Heads (single, time series, map)
  - Fluxes (single, time series, map)
- “Parameter”: model input
  - Every node, every stress period

Equation alert!

Modflow-6: Standard Equations for Adjoint Model

Forward model (MF6)
Solving for head (Forward in time)

$$\left[A\left(\vec{\alpha},\vec{h}^{k}\right)\right]\vec{h}^{k}=\vec{b}^{k}\left(\vec{\alpha},\vec{h}^{k-1}\right)$$
$$\vec{h}^{0}=\vec{h}_{i}$$



$$\left(\vec{h}^{1},\vec{h}^{2},\vec{h}^{3},\ldots\vec{h}^{N}\right)$$

Linear/Nonlinear

Performance measure

$$J=\sum_{k=1}^{N}F^{k}\left(\vec{\alpha},\vec{h}^{k}\right)$$

$$\frac{\partial J}{\partial \alpha_{i}}?$$

Modflow-6: Standard Equations for Adjoint Model

Forward model (MF6)
Solving for head (Forward in time)

$$[A(\vec{\alpha},\vec{h}^k)]\vec{h}^k = \vec{b}^k(\vec{\alpha},\vec{h}^{k-1})$$
$$\vec{h}^0 = \vec{h}_i$$

$$(\vec{h}^1,\vec{h}^2,\vec{h}^3,\dots\vec{h}^N)$$

Linear/Nonlinear

Adjoint model (MF6)
Solving for adjoint states (**Backward** in time)

$$[A(\vec{\alpha},\vec{h}^k)]^T\vec{\lambda}^{k-1} = -\left[\frac{\partial\vec{b}^{k+1}(\vec{\alpha},\vec{h}^k)}{\partial\vec{h}^k}\right]^T\cdot\vec{\lambda}^k - \frac{\partial F^k(\vec{\alpha},\vec{h}^k)}{\partial(\vec{h}^k)^T}$$
$$\vec{\lambda}^N = 0$$

$$(\vec{\lambda}^{N-1},\vec{\lambda}^{N-2},\vec{\lambda}^{N-3},\dots\vec{\lambda}^0)$$

Linear

Performance measure

$$J = \sum_{k=1}^{N} F^k(\vec{\alpha},\vec{h}^k)$$

$$\frac{\partial J}{\partial\alpha_i}?$$

Sensitivity coefficients

$$\frac{\partial J}{\partial\alpha_i} = \sum_{k=1}^{N}\left\{\underbrace{\frac{\partial F^k}{\partial\alpha_i}}_{\text{Direct effect}} + \underbrace{(\vec{\lambda}^{k-1})^T\left(\frac{\partial[A]}{\partial\alpha_i}\vec{h}^k - \frac{\partial\vec{b}^k}{\partial\alpha_i}\right)}_{\text{Indirect effect}}\right\}$$

What is the adjoint state conceptually…

- The sensitivity of a given performance measure to a unit injection of water in each active model cell for each solution time…

What is the adjoint state conceptually…

- The sensitivity of a given performance measure to a unit injection of water in each active model cell for each solution time…
- If your performance measure is simulated sw-gw exchange…
- …then the adjoint state is the capture fraction!

What is the adjoint state conceptually…

- The sensitivity of a given performance measure to a unit injection of water in each active model cell for each solution time…
- If your performance measure is simulated sw-gw exchange…
- …then the adjoint state is the capture fraction!

Design goals

- Non-intrusive: easy to maintain as MF6 advances
- Work with *almost* any MF6 GWF model
  - Newton and standard solution schemes
  - Structured and unstructured
- Wide range of performance measures
- Wide range of parameters
- Open source and free!

Approach

- Using the MF6 API
- Forward solve (linear/nonlinear):
  - Only done once
  - Advance thru simulation time, harvesting solution components
- Adjoint solve (always linear!):
  - One per performance measure
  - Solve for adjoint states backwards in time
  - Propagate to parameters via chain rule

Hughes and others (2022). The MODFLOW Application Programming Interface for simulation control and software interoperability

Implementation

- Mf6Adj class
  - Processes .adj file and prepares for forward solve
  - Solves the forward MF6 GWF model
  - Collects solution components
    - Conductance matrix, RHS, head, saturation
  - Writes solution info to HDF5
  - The only place the MF6 shared lib is needed!

Implementation

- PerfMeas class
  - Contains PerfMeasRecord instances
    - One per performance measure entry
    - Spatio-temporal information
    - Residual and weight info
  - Loads forward model HDF5 output file
  - Solves for adjoint state and propagates to parameters
  - Writes to (new) HDF5 file

Currently supported parameters/inputs

- Adjoint state (1 per node per stress period)
  - ~ well flux
- Recharge (1 per node per stress period)
  - area * adj state
- NPF k11,k33 (1 per node)
- STO ss (1 per node)
- GHB bhead,cond (1 per boundary node per stress period)
- DRN elev, cond (1 per boundary node per stress period)
- RIV stage, cond (1 per boundary node per stress period)
- SFR stage, cond (1 per reach per stress period)

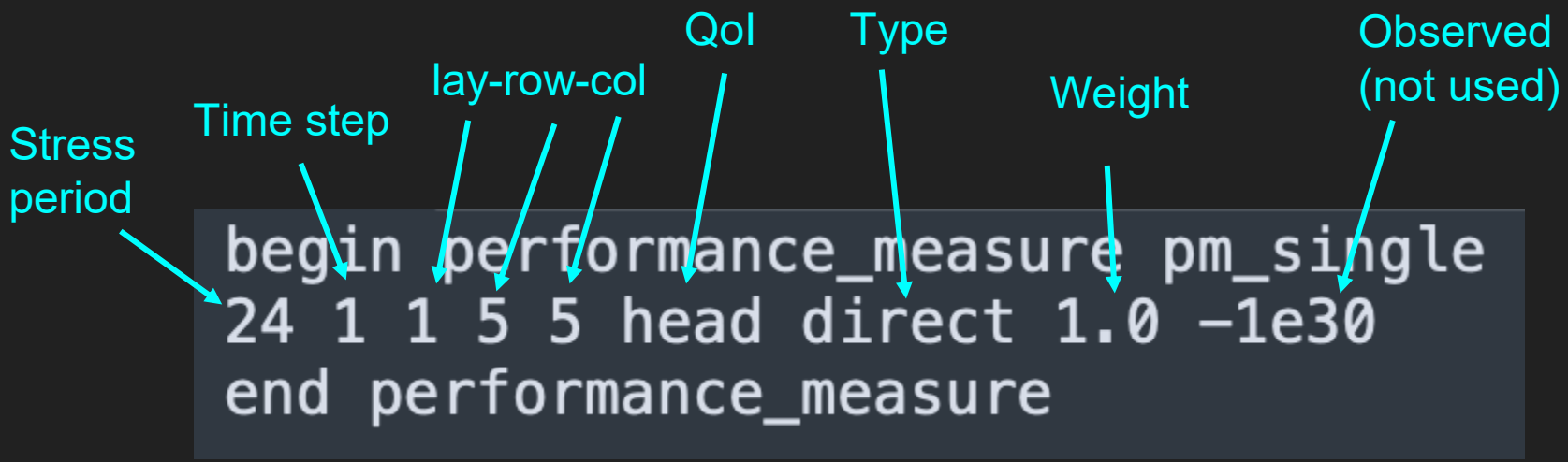
Currently supported performance measures

- Heads
  - At a point in space-time
  - A timeseries at one point in space
  - Aggregate over a region for one time
  - Combinations…
- Boundary fluxes
  - At a point in space-time
  - A timeseries at one point in space
  - Aggregate over a region for one time
  - Combinations…
- Sum of weighted head residuals
  - PEST-style objective function

Example performance measure blocks

```
begin performance_measure pm_single
24 1 1 5 5 head direct 1.0 -1e30
end performance_measure
```

Example performance measure blocks



Example performance measure blocks

```
begin performance_measure pm_single_alltimes
1 1 1 5 5 head direct 1.0 -1e30
2 1 1 5 5 head direct 1.0 -1e30
.
.
.
24 1 1 5 5 head direct 1.0 -1e30
25 1 1 5 5 head direct 1.0 -1e30
end performance_measure
```

Example performance measure blocks

```
begin performance_measure pm_ssr
1 1 3 3 10 head residual 1.0 34.597052432112854
1 1 3 10 2 head residual 1.0 34.61896038291642
.
.
.
11 1 1 27 7 head residual 1.0 36.428347809278l5
11 1 1 34 8 head residual 1.0 34.7015998123468 85
11 1 1 35 11 head residual 1.0 33.959077339551236
end performance_measure
```

Example performance measure blocks

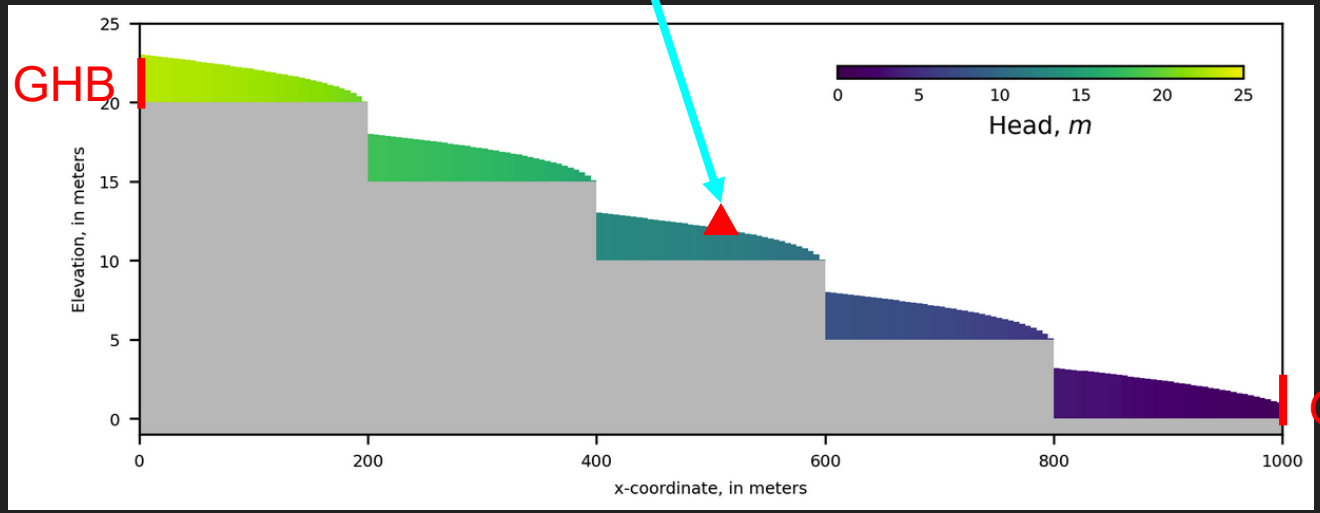
```
begin performance_measure headwater
23 1 1 1 16 sfr_1 direct 1.0 -1.0e+30
23 1 1 2 16 sfr_1 direct 1.0 -1.0e+30
.
.
.
23 1 1 20 16 sfr_1 direct 1.0 -1.0e+30
23 1 1 21 16 sfr_1 direct 1.0 -1.0e+30
end performance_measure
```

Usage

```
adj = mf6adj.Mf6Adj("test.adj",lib_name,verbose_level=1)
adj.solve_gwf()
adj.solve_adjoint()
adj.finalize()
```

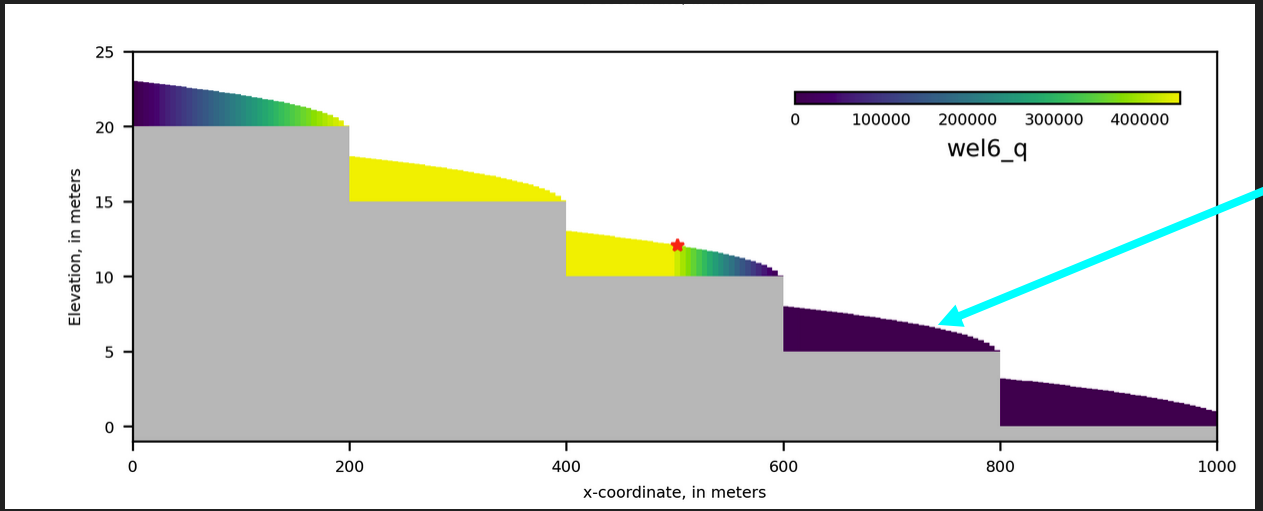
Zaidel problem

- Numerical benchmark for the newton solver
  - Part of the MODFLOW6 examples
- Performance measure: a single groundwater level



Zaidel problem

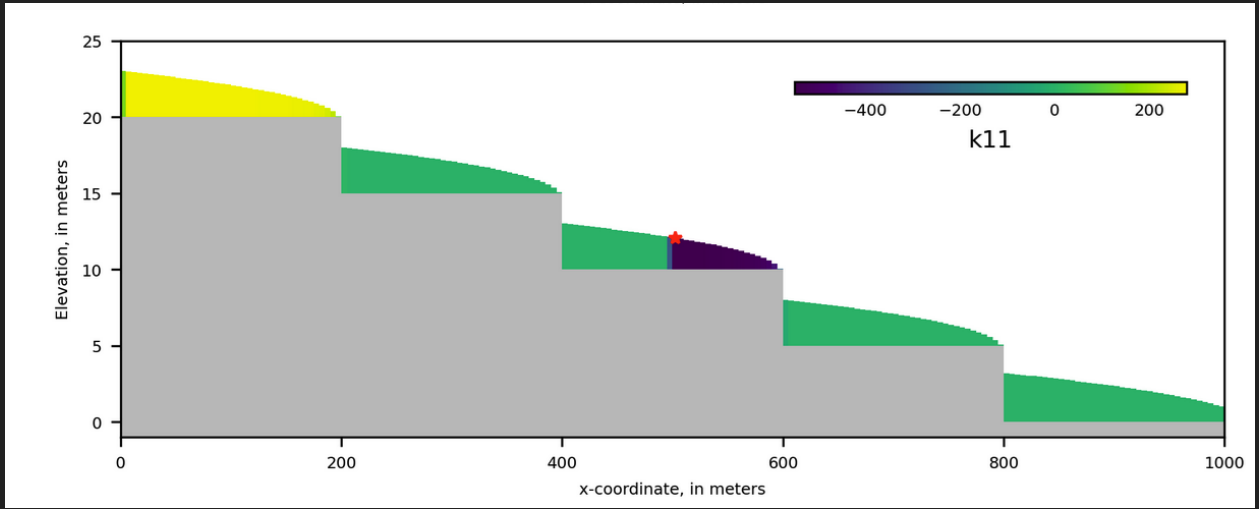
- Performance measure: a single groundwater level



Insensitive to pumping

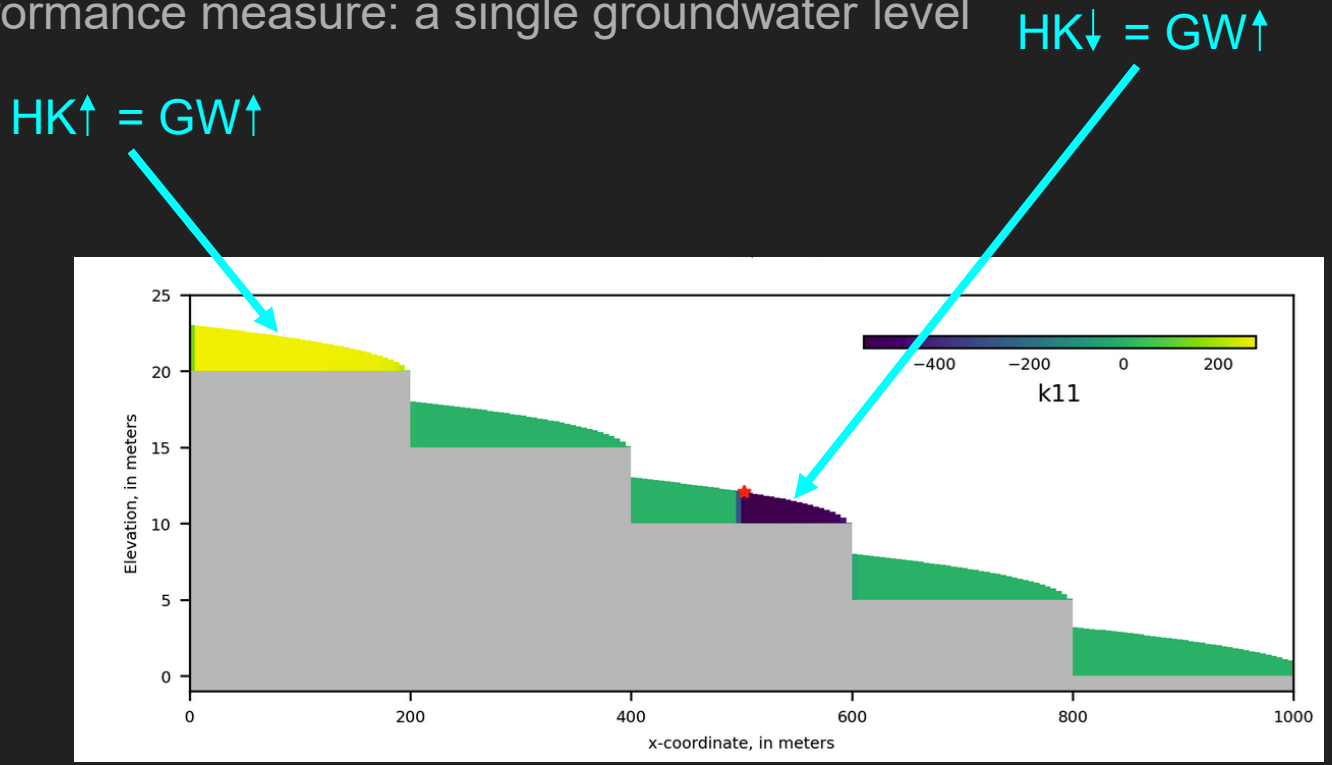
Zaidel problem

- Performance measure: a single groundwater level



Zaidel problem

- Performance measure: a single groundwater level



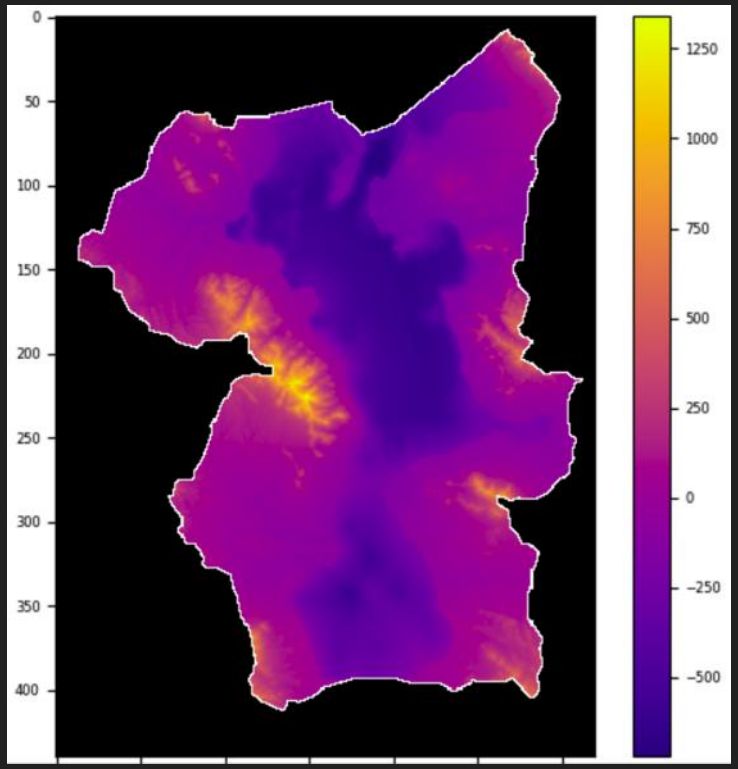
Example: San Pedro

A New Capture Fraction Method to Map How Pumpage Affects Surface Water Flow

Stanley A. Leake, Howard W. Reeves, Jesse E. Dickinson

First published: 19 August 2010 | https://doi.org/10.1111/j.1745-6584.2010.00701.x | Citations: 29

- Originally developed by Stan Leake
- Focus on simulating the relation between gw use and sw flow
- Structured grid, transient
- Thanks to Joe Hughes for porting to MF6!
- An example notebook in the repo…



Example: San Pedro

A New Capture Fraction Method to Map How Pumpage Affects Surface Water Flow

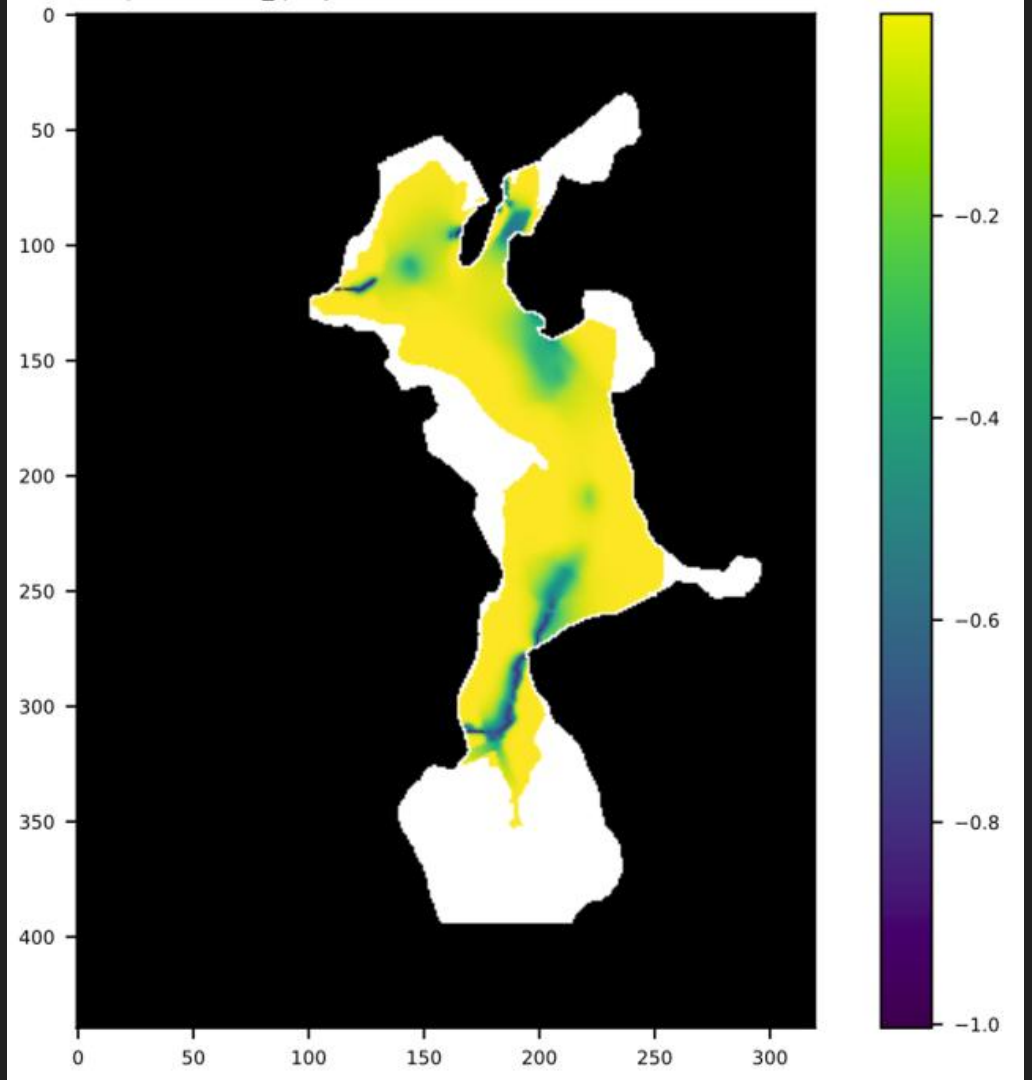
Stanley A. Leake ✉, Howard W. Reeves, Jesse E. Dickinson

First published: 19 August 2010 | https://doi.org/10.1111/j.1745-6584.2010.00701.x | Citations: 29

- Performance measure:
  - Sum of SW-GW exchange over all reaches
- 1 forward mf6 run then 1 adjoint run
- ~115K active nodes…~600K model runs required for perturbation sensitivities

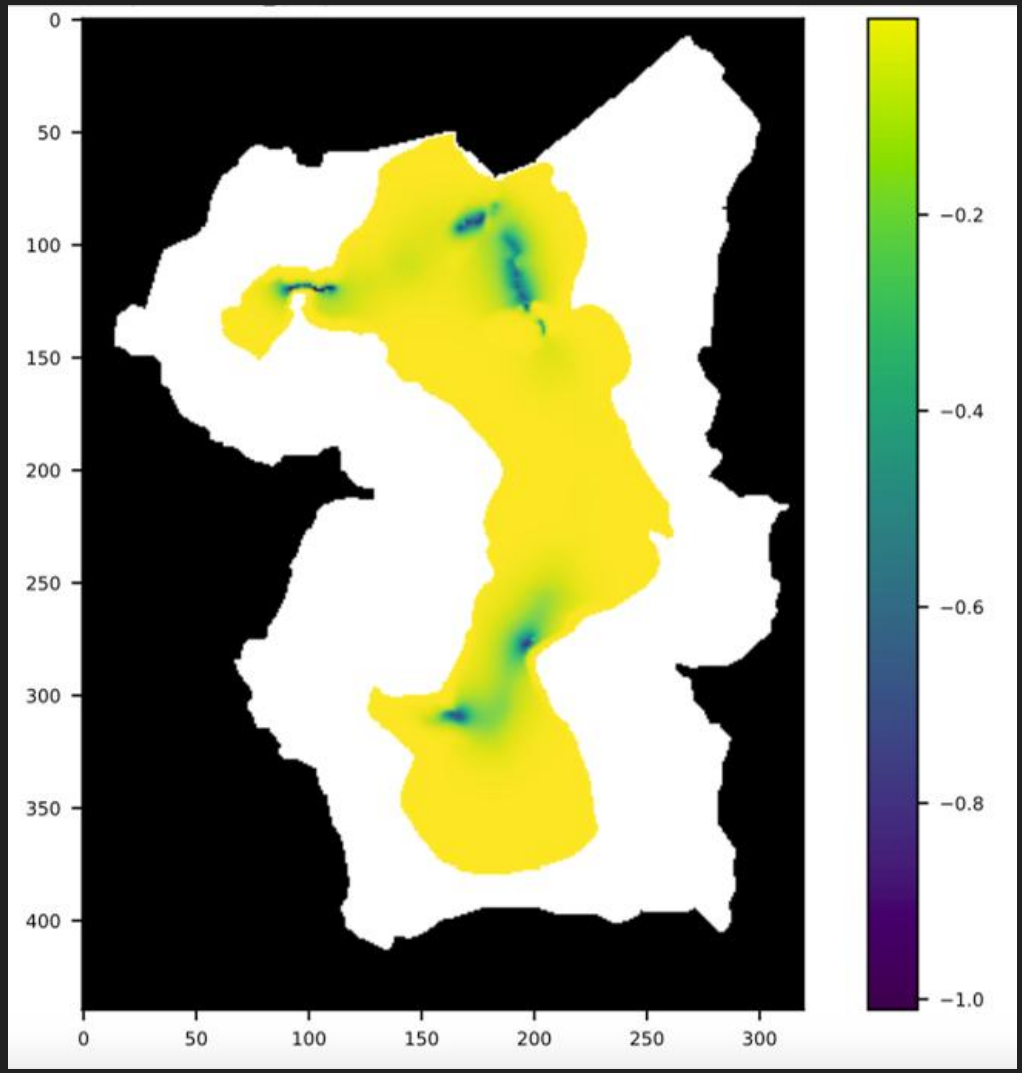
Example: San Pedro

- Capture fraction layer 4
- Higher values = more depletion



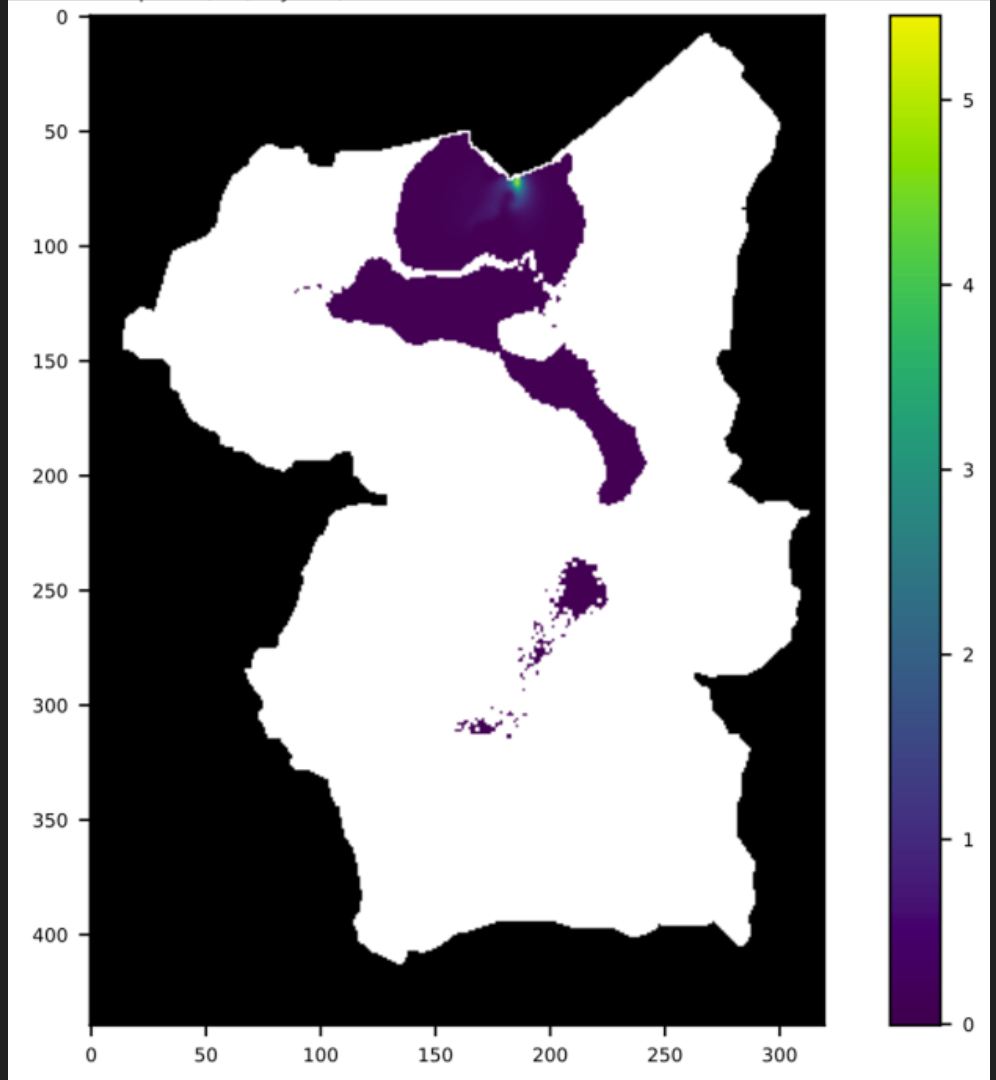
Example: San Pedro

- Capture fraction layer 5
- Higher values = more depletion



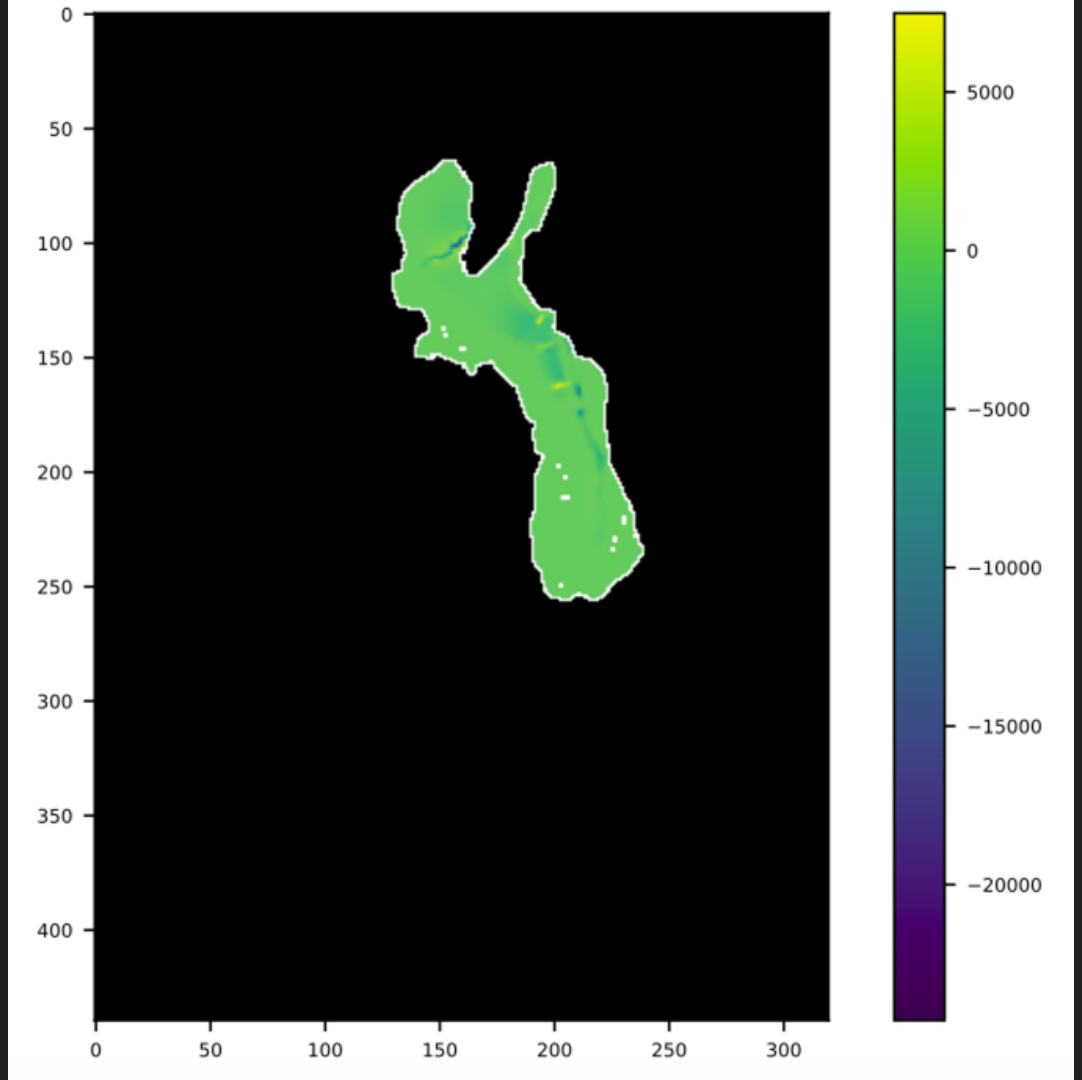
Example: San Pedro

- SS layer 5



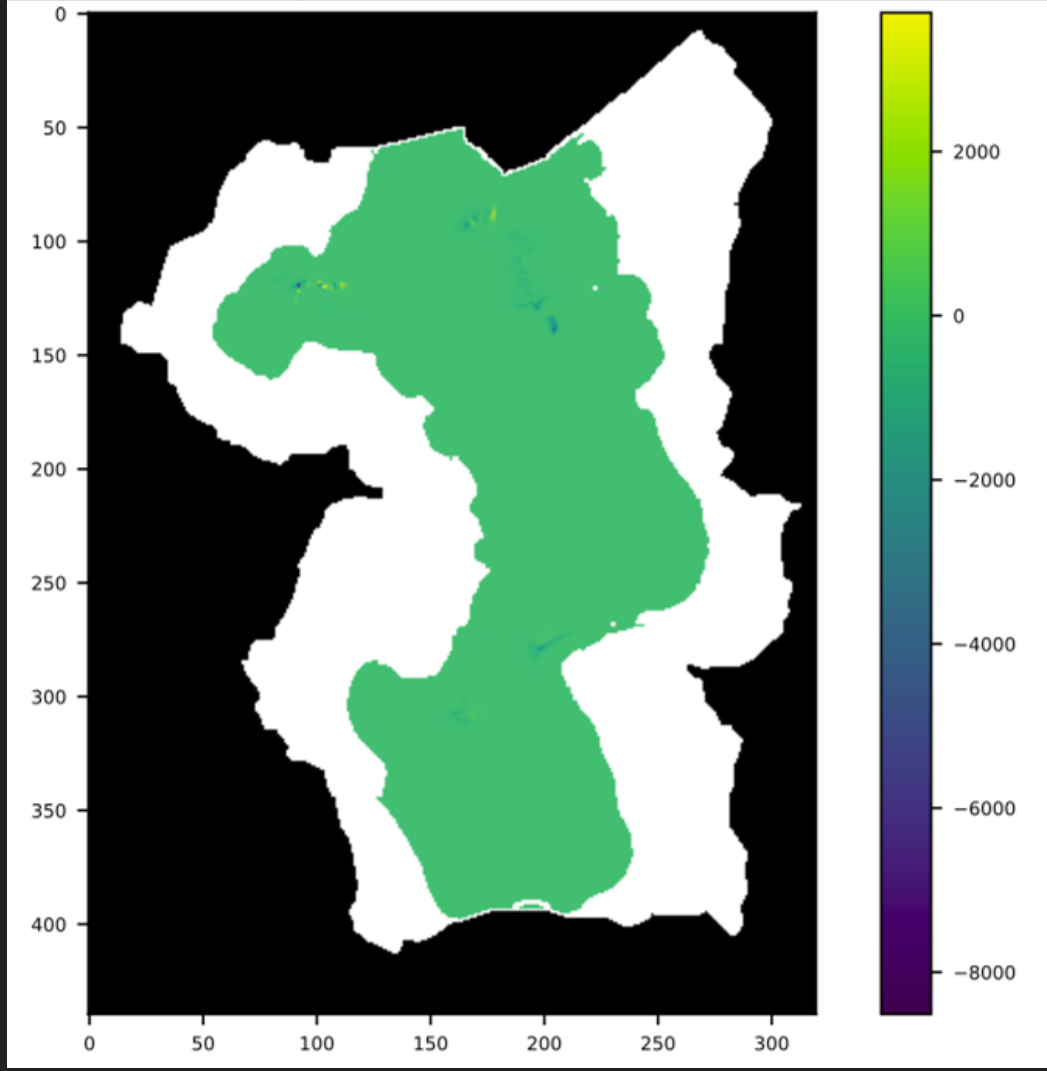
Example: San Pedro

- VK layer 3



Example: San Pedro

- HK layer 5



Future directions

- Quasi-newton optimization
  - Optimization
  - RML
- Hamiltonian Monte Carlo
  - Efficient metropolis sampling - needs gradients
- Model emulator training

Thanks

Contributions welcome!

https://github.com/INTERA-Inc/mf6adj