Deltares

enabling delta life

Use case BMI/XMI
reducing computational time with MODFLOW 6

Hendrik Kok

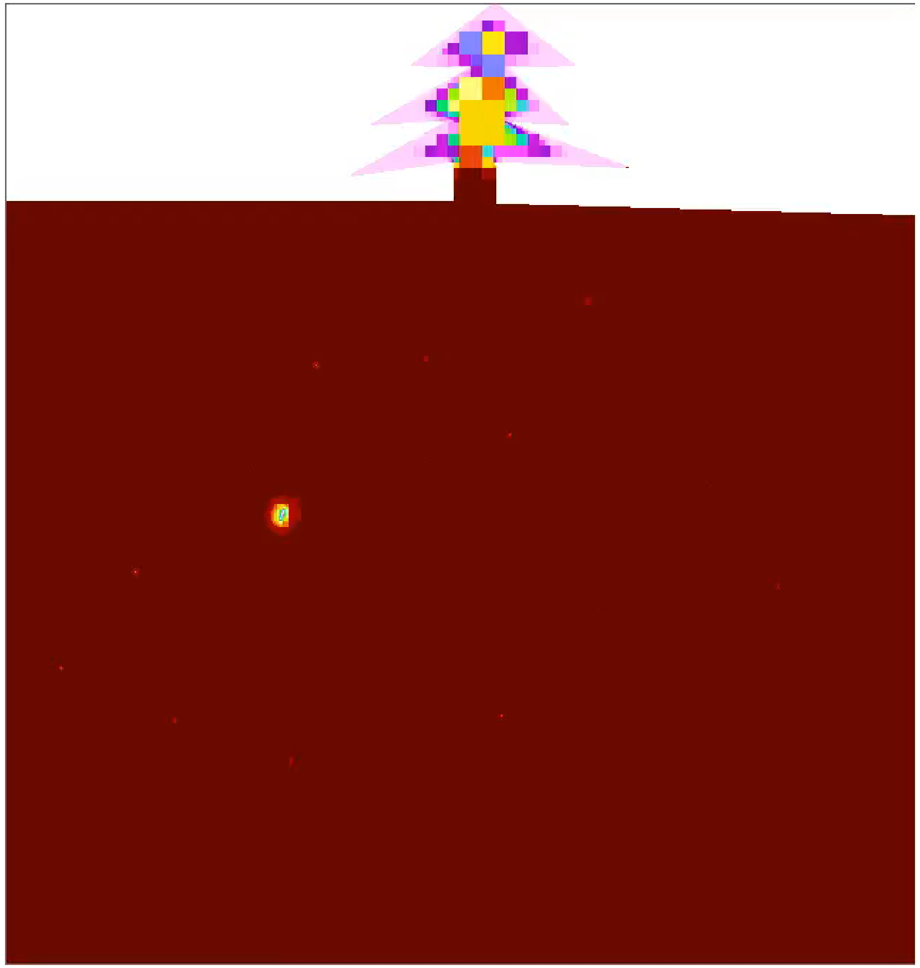
Andre Blonk

Huite Bootsma

28 november 2019

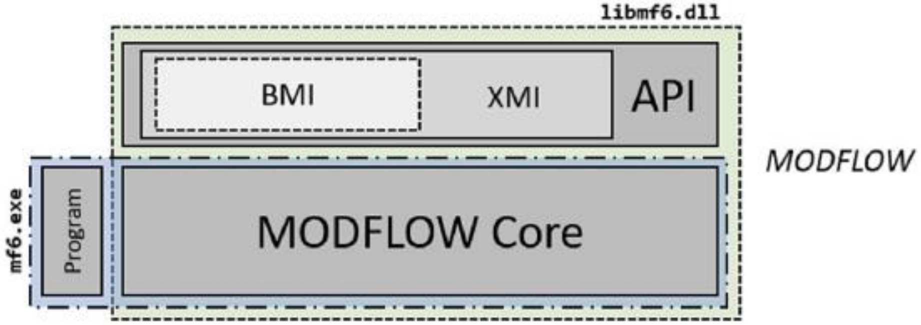
Content

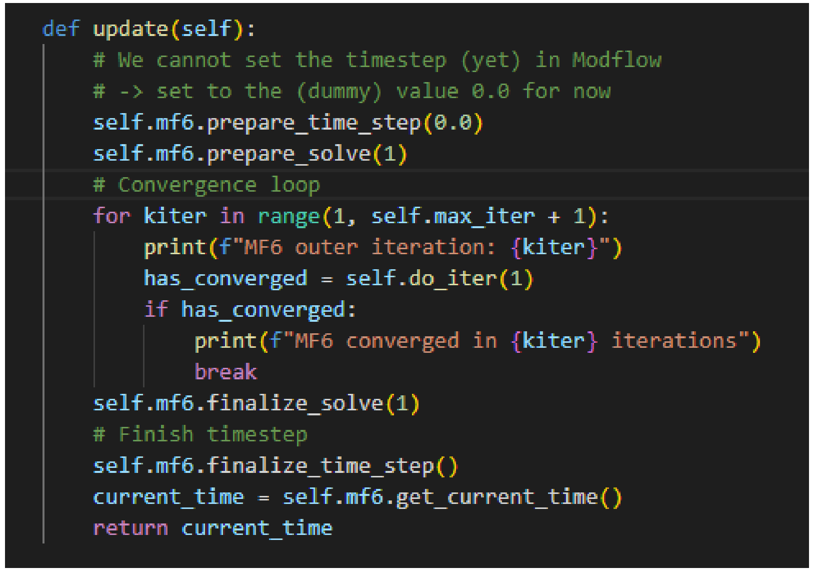
- BMI/XMI
- Testcase:
  - Problem definition
  - Possible solutions
  - Adapted solution
- Results
- Release



BMI/XMI background

- Communication between (computational) kernels is controlled via the Application Programming Interface (API)
- BMI/XMI are a set of conventions on how to control simulations so it can easily be controlled from external source by:
  - A standardized set of calls like ‘initialize model’, ‘prepare timestep’ and ‘finalize timestep’
  - Internal data exchange via ‘get_value’ functionality
- Kernels are shared as dynamic libraries (.dll) when using BMI/XMI.
  - Individual kernels can be continuously updated
  - Kernel needs to be controlled externally via ‘driver’

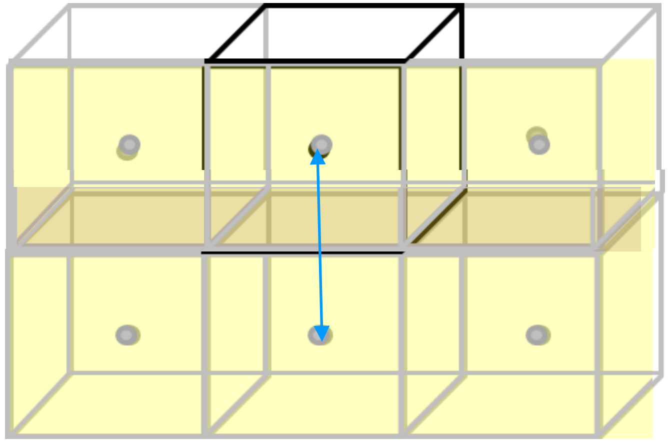


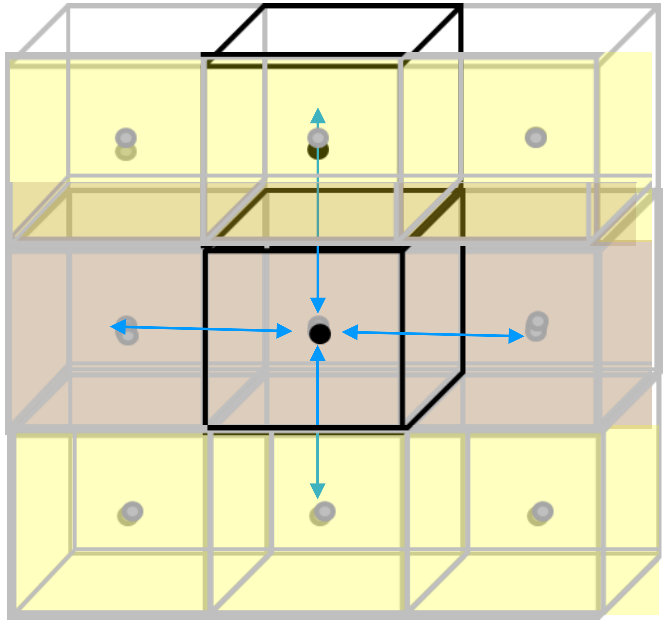


```python
def update(self):
    # We cannot set the timestep (yet) in Modflow
    # -> set to the (dummy) value 0.0 for now
    self.mf6.prepare_time_step(0.0)
    self.mf6.prepare_solve(1)
    # Convergence loop
    for kiter in range(1, self.max_iter + 1):
        print(f"MF6 outer iteration: {kiter}")
        has_converged = self.do_iter(1)
        if has_converged:
            print(f"MF6 converged in {kiter} iterations")
            break
    self.mf6.finalize_solve(1)
    # Finish timestep
    self.mf6.finalize_time_step()
    current_time = self.mf6.get_current_time()
    return current_time
```

Testcase

- MODFLOW 2005 supports ‘quasi-3D’ discretisation of the subsoil
  - Subsoil parametrization based on K_h and C
  - Aquitards are not in the schematisation, presence is discounted in leakage tem
- MODFLOW 6 only supports ‘full 3D’ discretisation of the subsoil
  - Subsoil parametrization based on K_h and K_v
  - Aquitards need to be included in subsoil schematization (when present)
- For large regional models this means roughly doubling the computational time (nlay-1 times)





Deltares

Possible solution

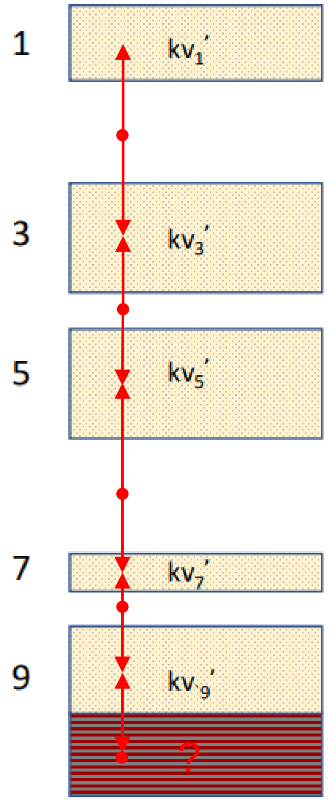
Move contribution K_v of aquitards to aquifers (K_v')

- K_v' then includes the contribution from top and bottom resistance
- K_v' can be calculated from a set of equations, when using a dummy value for bottom K_v'

This methods works but:

1. Results most of the time in negative K_v' values, which are not excepted as model input
2. Only possible with DISU because of non continues top and bots

Therefore, changes in source code needed



Deltares

Adapted solution

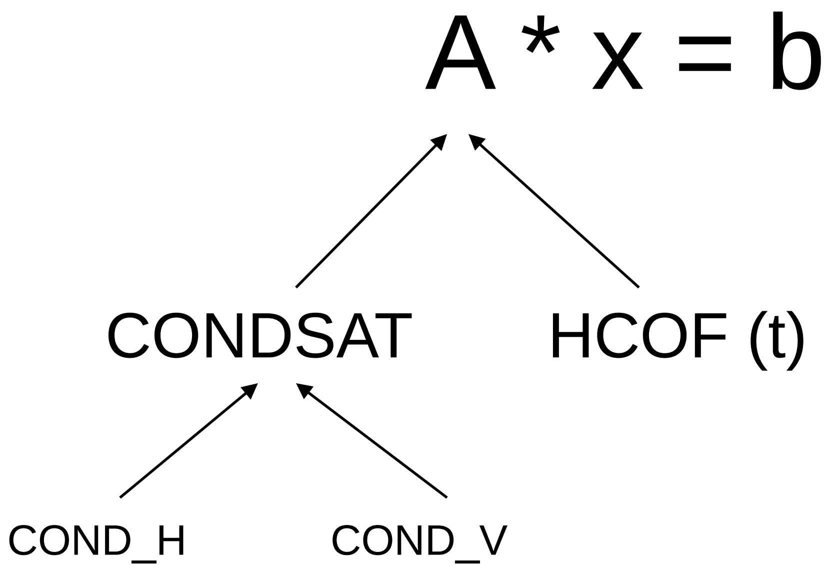
Use BMI/XMI implementation to recalculated conductance after initialization

‘CONDSAT’ array stores all cell connection data, which is updated to coefficient matrix per timestep

- Vertical conductance not updated during simulation when:
  - No VARAIBLECV is used

CONDSAT

- Sparse array and only upper diagonal is stored (n -> m is equal to m -> n)
- Includes both horizontal and vertical node-to-node connections



A * x = b

CONDSAT

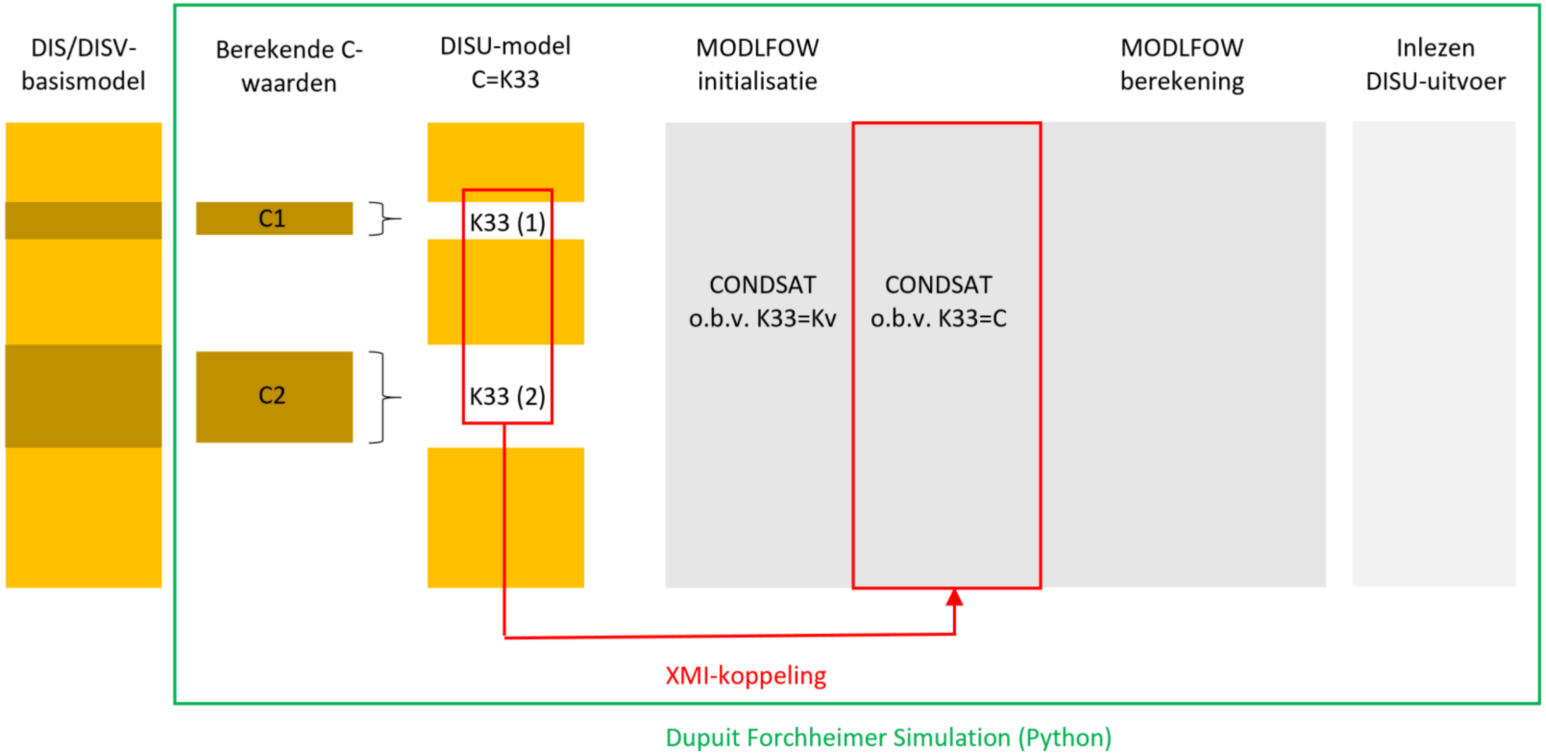
HCOF (t)

COND_H

COND_V

Adapted solution

Python class ‘DupuitForchheimerSimulation’ to implement k33 as vertical resistance (days)

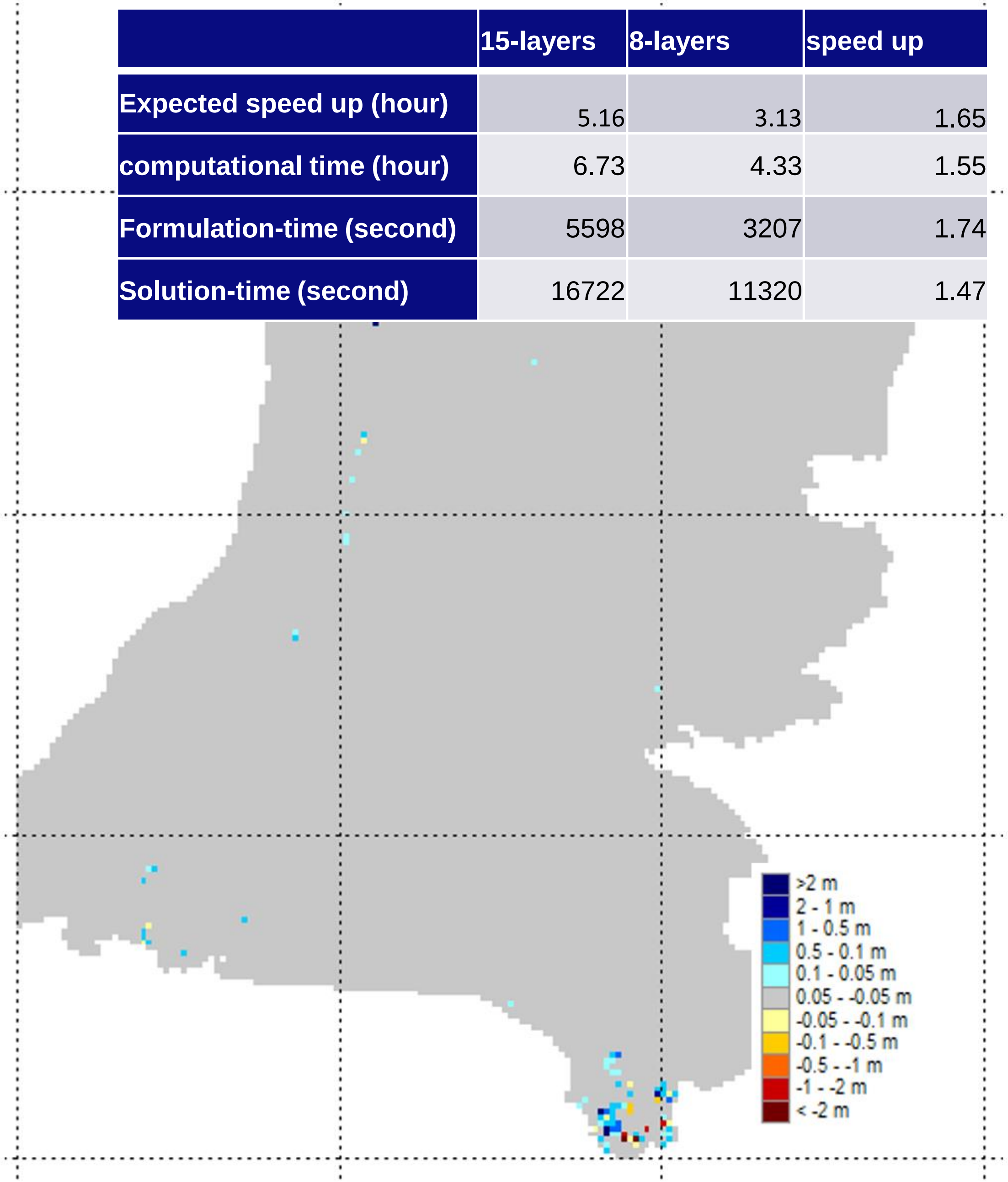


Deltares

Results

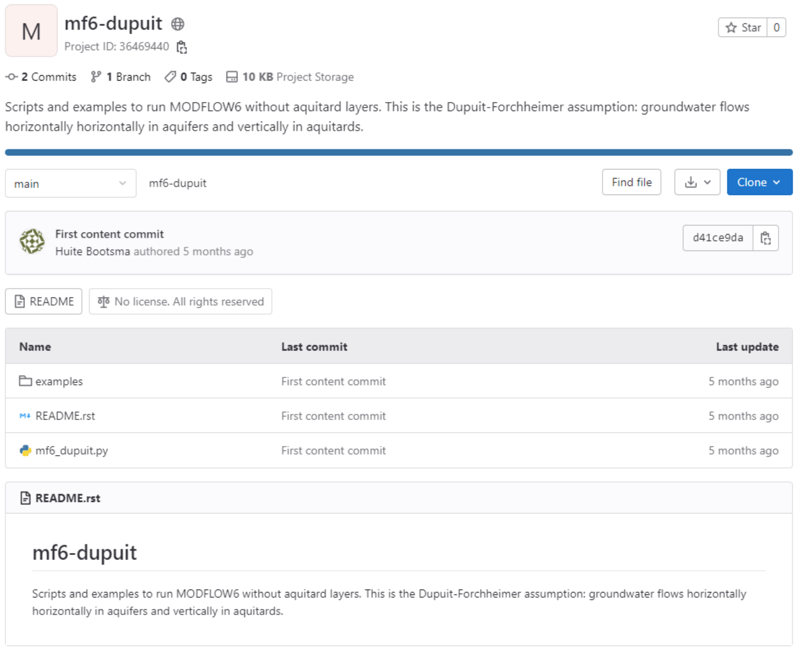
- Speedup close to expected values
  - Based on speed decrease form migration from MF2005 to MF6
- BMI/XMI are a nice set of tools to intervene in MF6-simulation
- When using BMI/XM, (extensive) knowledge of internal structure of computational kernel is required

	15-layers	8-layers	speed up
Expected speed up (hour)	5.16	3.13	1.65
computational time (hour)	6.73	4.33	1.55
Formulation-time (second)	5598	3207	1.74
Solution-time (second)	16722	11320	1.47



Release

[Deltares / iMOD / mf6-dupuit · GitLab]



M **mf6-dupuit**

Project ID: 36469440

☆ Star 0

2 Commits **1** Branch **0** Tags **10 KB** Project Storage

Scripts and examples to run MODFLOW6 without aquitard layers. This is the Dupuit-Forchheimer assumption: groundwater flows horizontally horizontally in aquifers and vertically in aquifers.

main mf6-dupuit Find file Clone

First content commit
Huite Bootsma authored 5 months ago
d41ce9da

README No license. All rights reserved

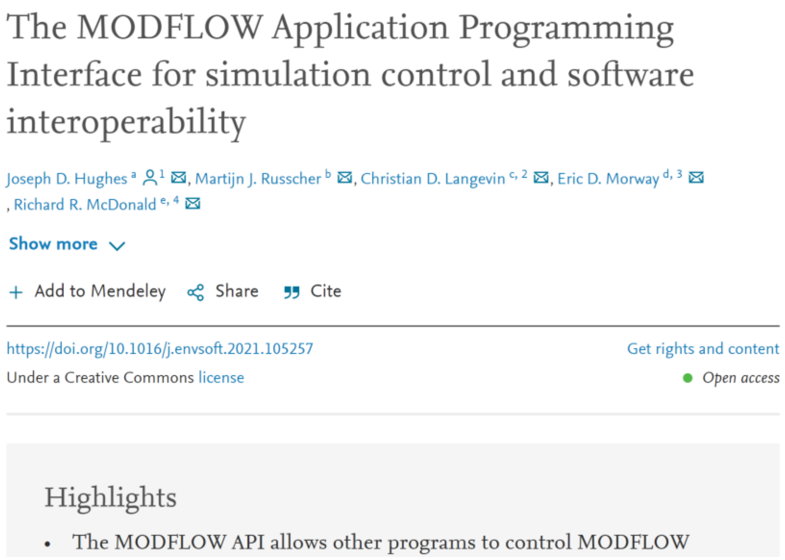
Name	Last commit	Last update
examples	First content commit	5 months ago
README.rst	First content commit	5 months ago
mf6_dupuit.py	First content commit	5 months ago

README.rst

mf6-dupuit

Scripts and examples to run MODFLOW6 without aquitard layers. This is the Dupuit-Forchheimer assumption: groundwater flows horizontally horizontally in aquifers and vertically in aquitards.

[The MODFLOW Application Programming Interface for simulation control and software interoperability - ScienceDirect]



The MODFLOW Application Programming Interface for simulation control and software interoperability

Joseph D. Hughes [a] [1], Martijn J. Russcher [b], Christian D. Langevin [c, 2], Eric D. Morway [d, 3], Richard R. McDonald [e, 4]

Show more

+ Add to Mendeley Share Cite

https://doi.org/10.1016/j.envsoft.2021.105257 Get rights and content

Under a Creative Commons license ● *Open access*

Highlights

- The MODFLOW API allows other programs to control MODFLOW

Contact

www.deltares.nl

@deltares

linkedin.com/company/deltares

info@deltares.nl

@deltares

facebook.com/deltaresNL