Parallel MODFLOW 6

By the MODFLOW Development Team

MODFLOW 6 and FloPy: Take Your Modeling to the Next Level

Princeton, New Jersey, USA

Friday, May 31 – June 1, 2024

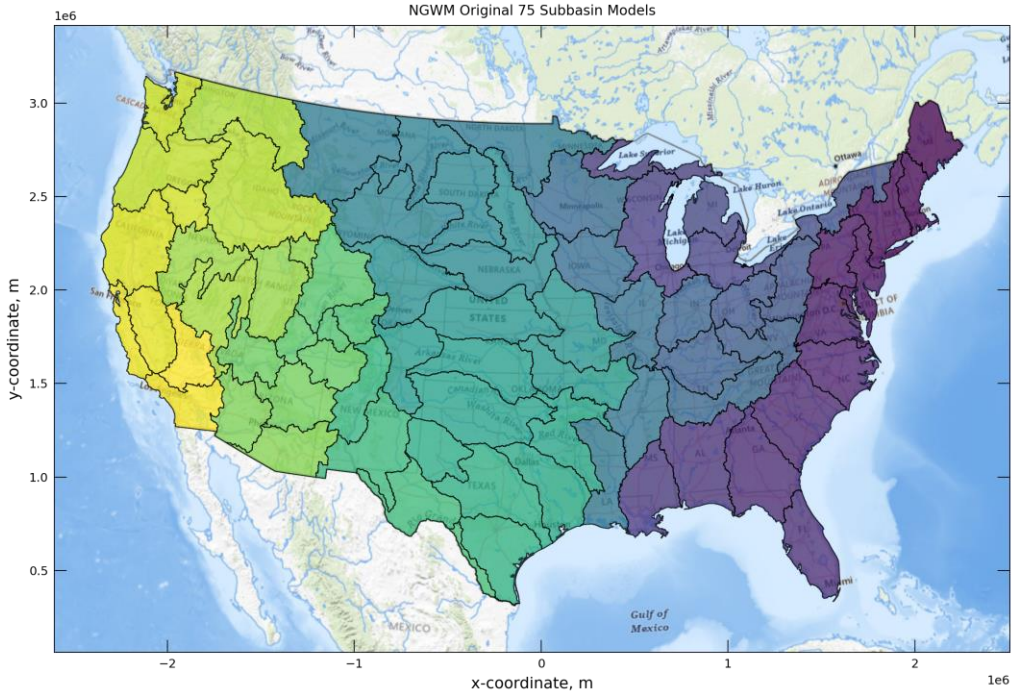
USGS science for a changing world

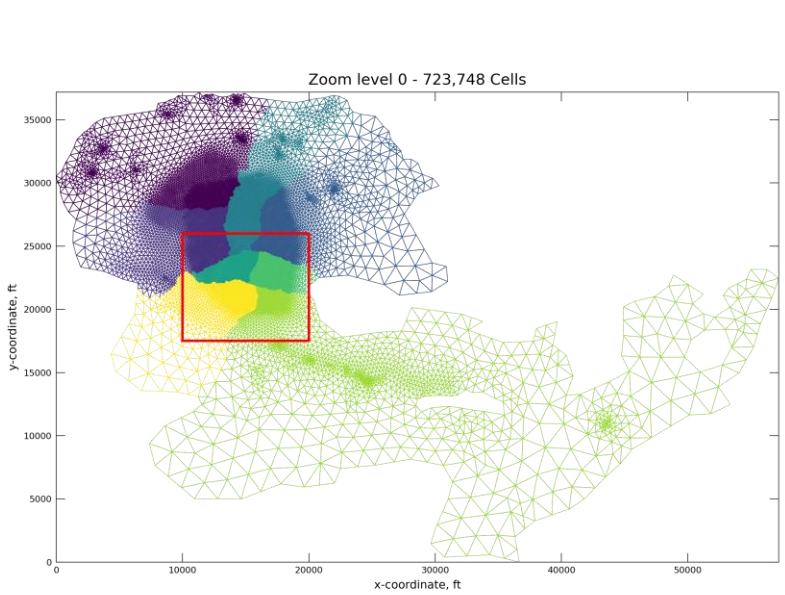
Deltares

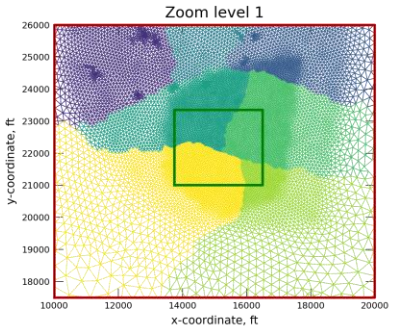
GSI ENVIRONMENTAL

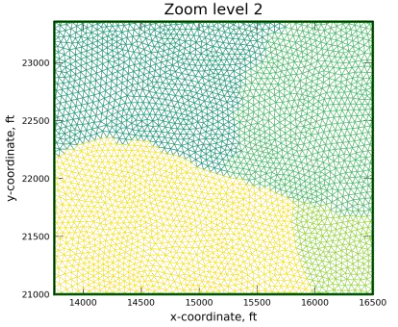
UCAR

Motivation









Model of a Regional Groundwater Basin in Southern California, Tetra Tech 2023

- Large national/continental scale models
- High resolution, regional models

Moore’s Law

Cramming More Components onto Integrated Circuits

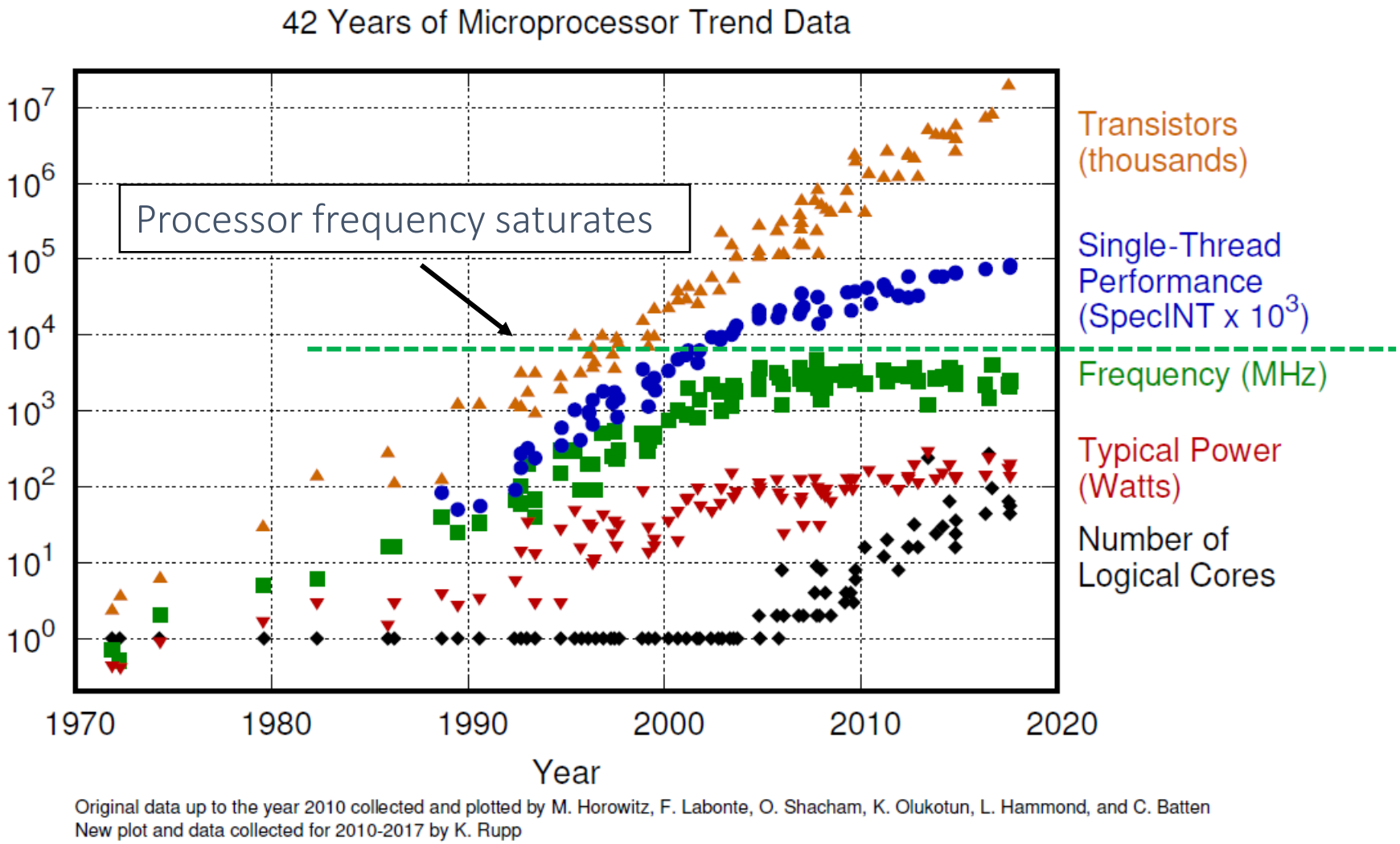
GORDON E. MOORE, LIFE FELLOW, IEEE

Integrated circuits will lead to such wonders as home computers—or at least terminals connected to a central computer—automatic controls for automobiles, and personal portable communications equipment. The electronic wristwatch needs only a display to be feasible today.

from *Electronics*, pp. 114–117, April 19, 1965

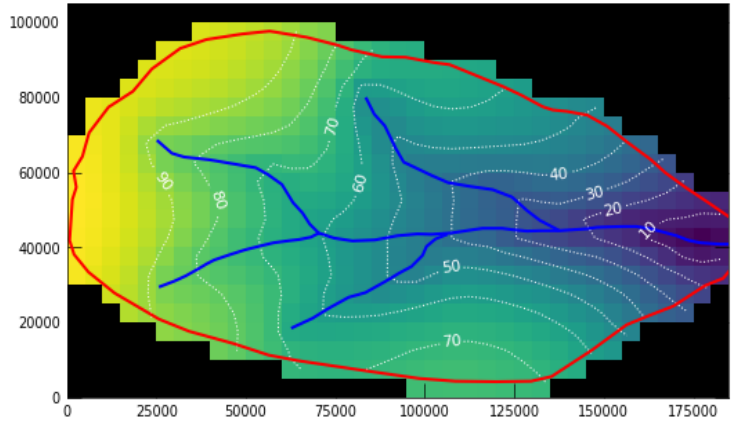
Gordon Moore (1929-2023)

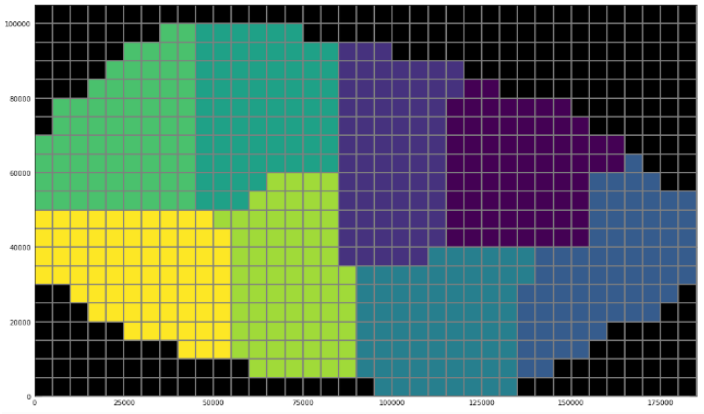
Moore's Law: the end of it…



Strategy: Domain Decomposition

- Instead of running a model on 1 processor core …
- … distribute over many cores
- Exchange of data at the domain interface
- Efficiency affected by
  - Amount of communication between domains
  - Load imbalance: solution of multiple domains proceeds in lock step…





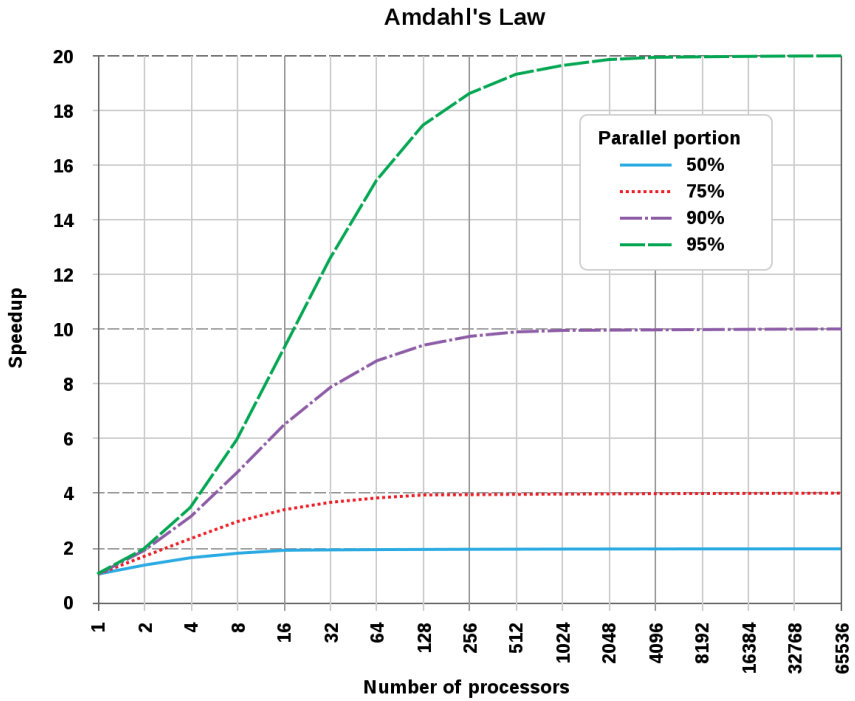
Limit on Speedup

- Further division of the work (i.e. smaller domains, increase in processor cores) increases communication overhead
- Put differently: how large can you make your model?!

- Amdahl's law:

$$\text{Speedup} \leq \frac{1}{1 - p}$$

with p = parallel portion



Parallel Solution

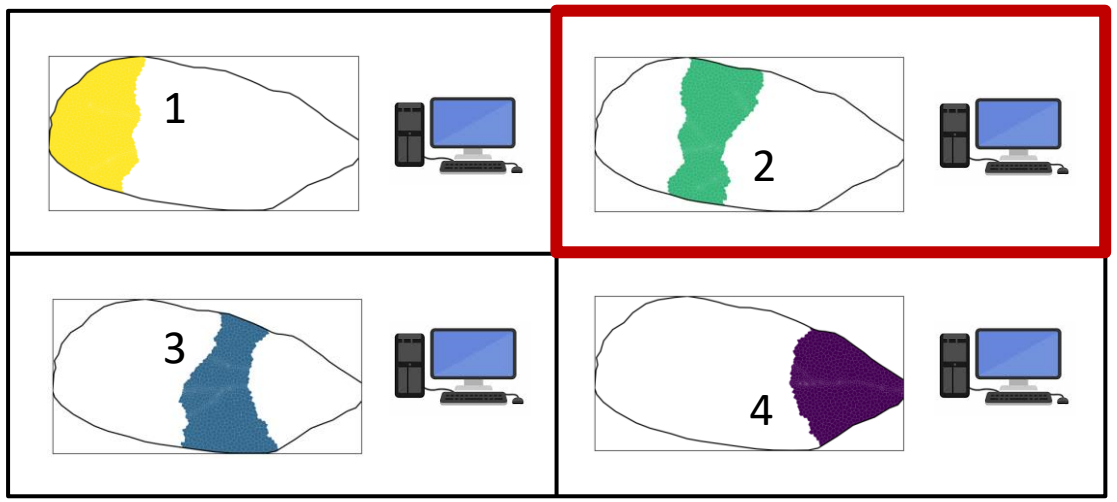
- Example: solve the groundwater flow equation (Ax = b) over 4 domains
- Dependent variable **x** (head) and right-hand side **b** are distributed
- Each process now manages a contiguous set of rows of the system matrix

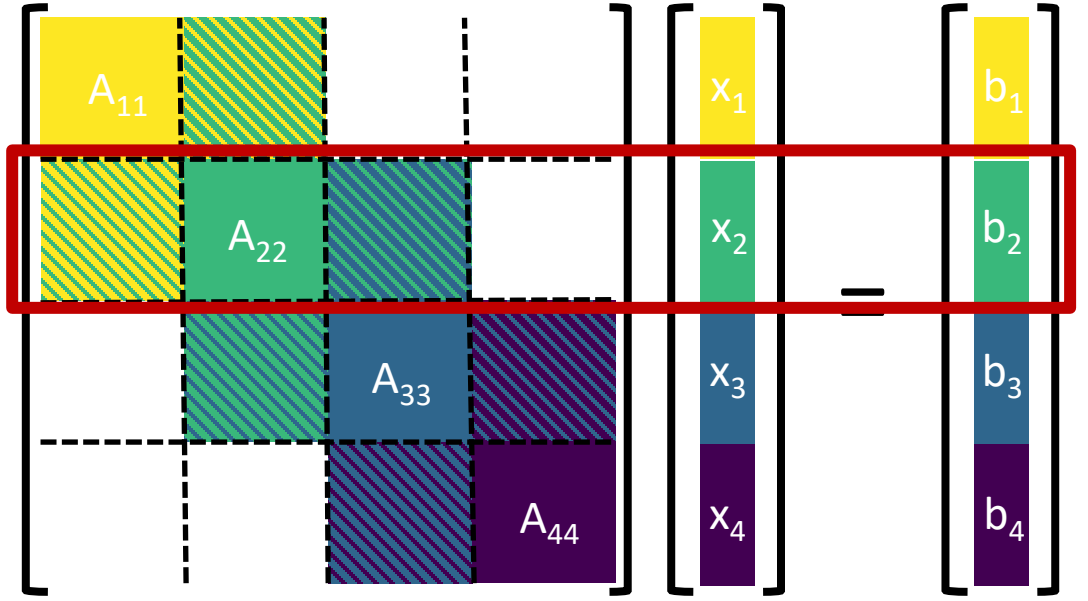
To solve this system:

- choose A = M – N
- M ≈ A and M_{ij}
- M_{ij} = 0 for i ≠ j (Block-Jacobi), we use the iterative scheme:

$$\mathbf{x}_i^{k+1} = M_{ii}^{-1} (\Sigma_j N_{ij} \mathbf{x}_j^k + \mathbf{b}_i)$$

- Does not depend on $\mathbf{x}_j^{k+1}$, hence parallel



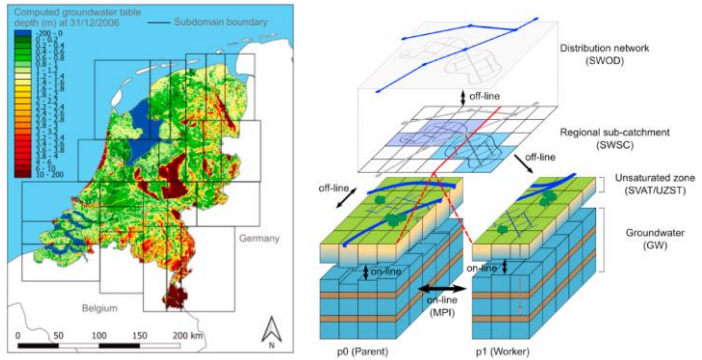


Earlier Work

- Good results for national-scale model
- Global groundwater with MODFLOW 6 prototype

More work needed: a **framework solution**

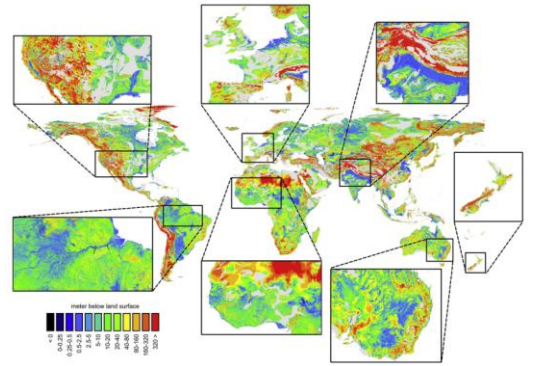
- Current and future model types
- Feature complete: XT3D, BUY, TVD, MVR, …
- Possible to add or modify hydrologic concepts without expert knowledge on parallelization
- MODFLOW remains available in its core set of functionality
- FloPy support for pre- and post-processing



Distributed memory parallel groundwater modeling for the Netherlands Hydrological Instrument

J. Verkaik[a,b,*], J.D. Hughes[c], P.E.V. van Walsum[d], G.H.P. Oude Essink[a,b], H.X. Lin[e,f], M.F. P. Bierkens[b,a]

[a] Unit Subsurface and Groundwater Systems, Deltares, Utrecht, the Netherlands
[b] Department of Physical Geography, Utrecht University, Utrecht, the Netherlands
[c] U.S. Geological Survey Integrated Modeling and Prediction Division, Chicago, United States
[d] Wageningen Environmental Research, Wageningen, the Netherlands
[e] Delft Institute of Applied Mathematics, Delft University of Technology, Delft, the Netherlands
[f] Institute of Environmental Sciences, Leiden University, Leiden, the Netherlands



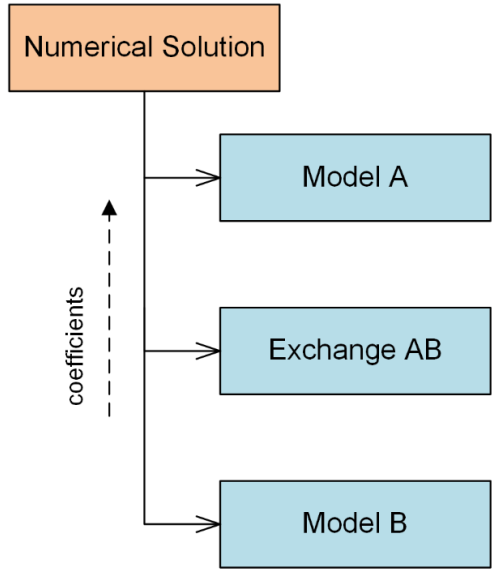
GLOBGM v1.0: a parallel implementation of a 30 arcsec PCR-GLOBWB-MODFLOW global-scale groundwater model

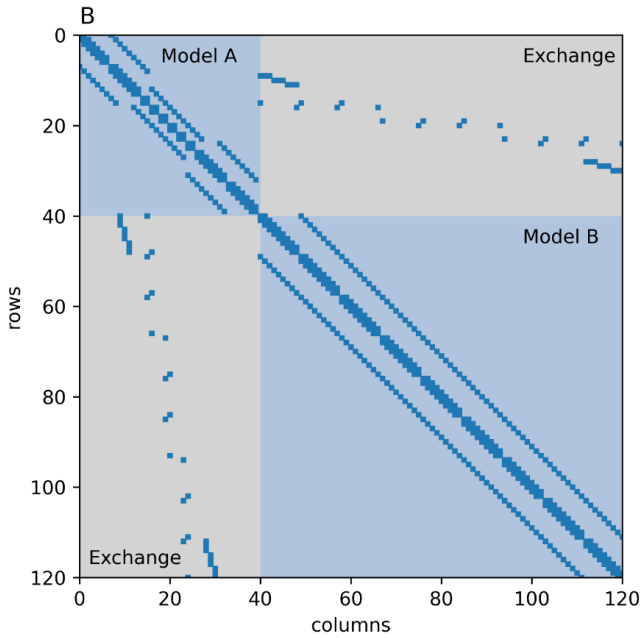
Jarno Verkaik[1,2], Edwin H. Sutanudjaja[2], Gualbert H. P. Oude Essink[1,2], Hai Xiang Lin[3,4], and Marc F. P. Bierkens[2,1]

[1]Unit Subsurface and Groundwater Systems, Deltares, Utrecht, the Netherlands
[2]Department of Physical Geography, Faculty of Geosciences, Utrecht University, Utrecht, the Netherlands
[3]Department of Applied Mathematical Analysis, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, the Netherlands
[4]Institute of Environmental Sciences, Faculty of Science, Leiden University, Leiden, the Netherlands

Framework Solution

- MODFLOW 6 has conceptualized a hydrologic process model into a “Numerical Model”
- Multiple models can be connected with Exchanges to be solved in a single Numerical Solution





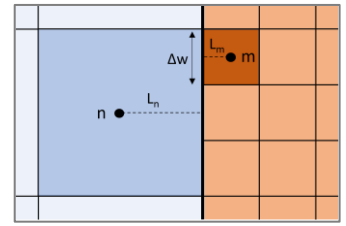
Framework Solution

Building Blocks:

1. Generalized coupling of Models based on the Exchanges
2. Virtualization of Data with synchronization: MPI
3. Parallel solution: PETSc

PETSc

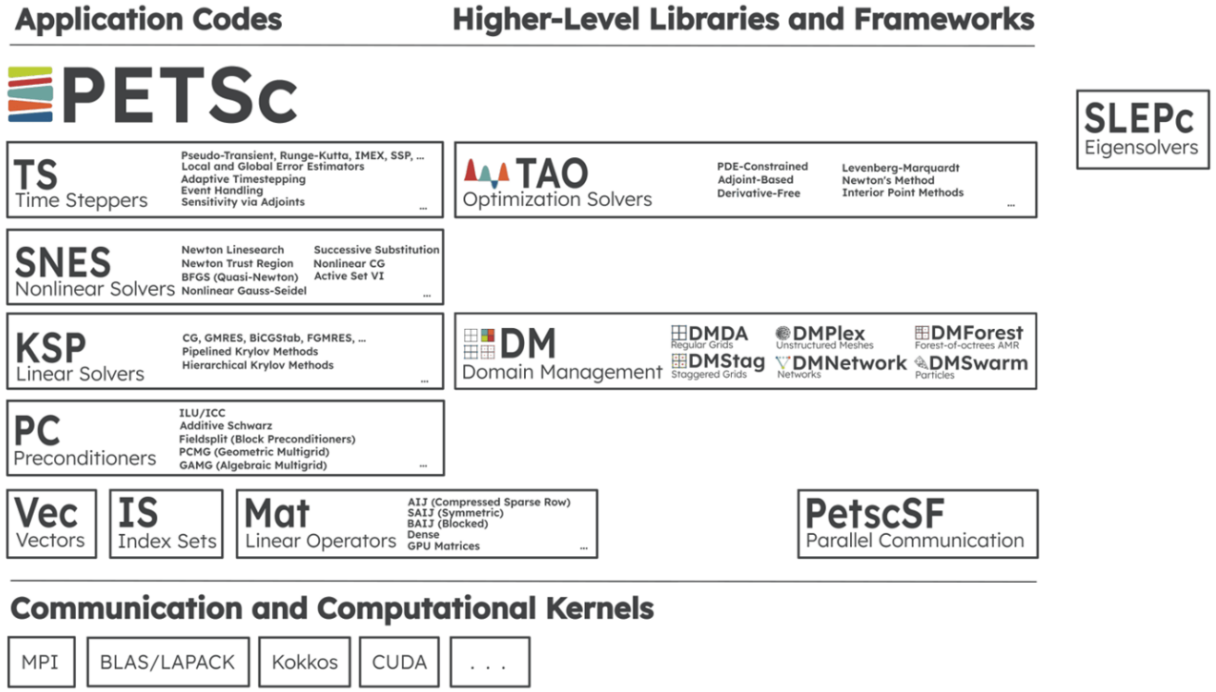
```
# File generated by Flopy ...
BEGIN options
PRINT_INPUT
END options

BEGIN dimensions
NEXG  36
END dimensions

BEGIN exchangedata
1 3 2  1 1 1  1  50.00000000  16.66666667  33.33333333
1 3 2  1 2 1  1  50.00000000  16.66666667  33.33333333
...
END exchangedata
```



More on this: Monday, 15:20

Parallel Solver: PETSc

- The Portable, Extensible Toolkit for Scientific Computation
- Suite of data structures and routines developed and maintained by ANL
- Scalable (parallel) solution of scientific applications modeled by partial differential equations
- Robust, Efficient, Portable
- Fortran bindings
- And: PETSc solver configured from IMS settings, including MILU(T)

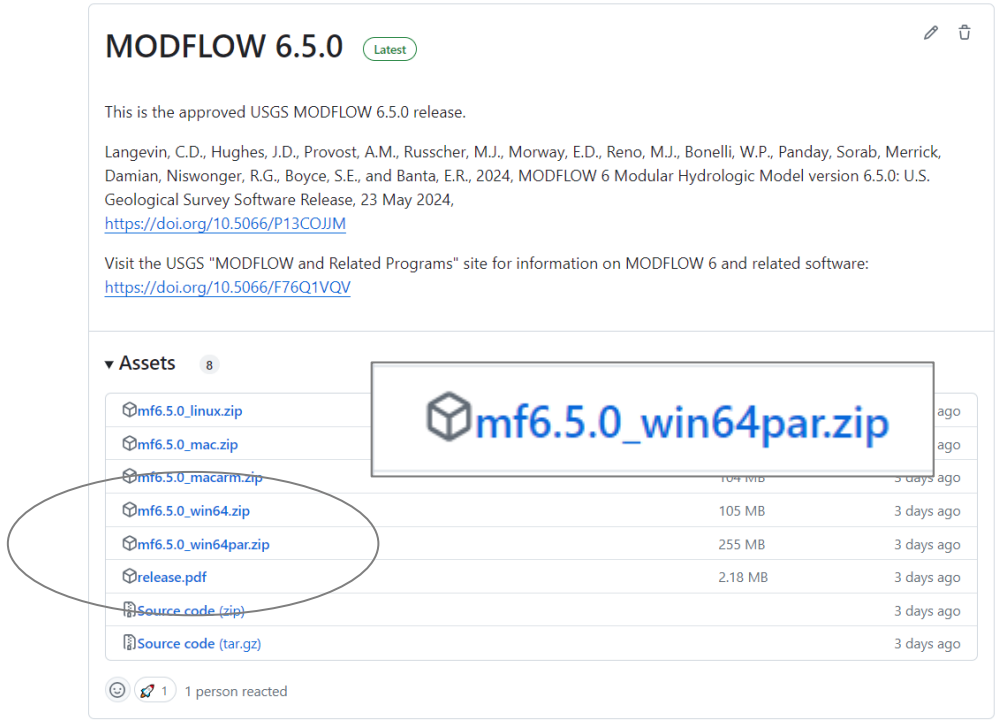


How to get it?

Windows:

- Download ‘win64par.zip’
- Install through FloPy:

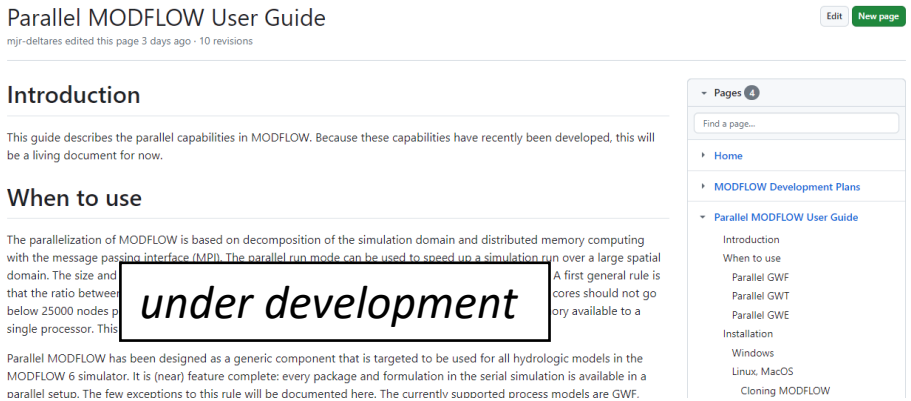
```
get-modflow --repo modflow6 --ostag win64par :python
```



Linux/Mac

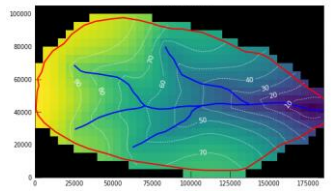
- Instructions in the parallel user guide:

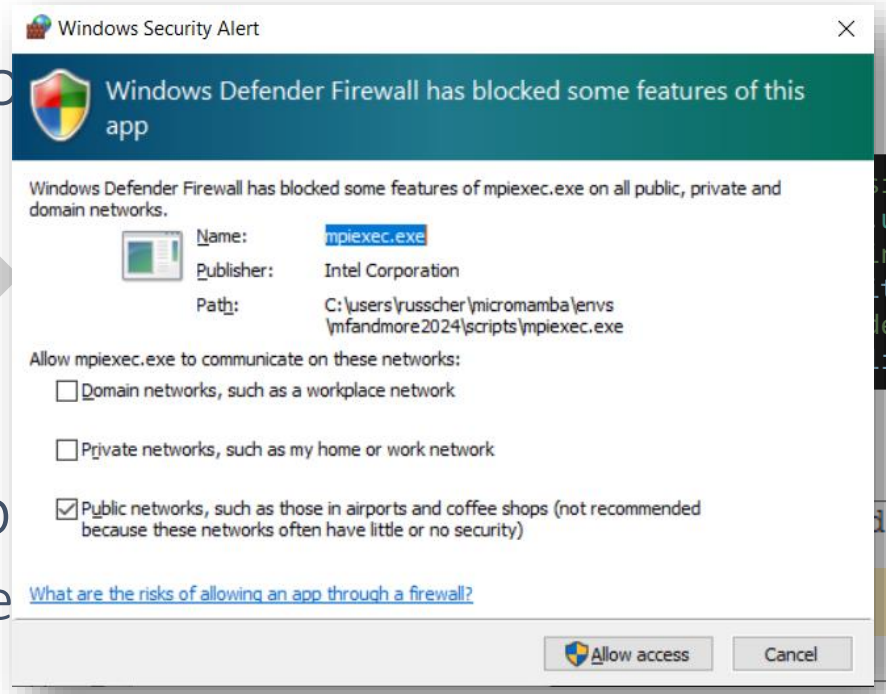
https://github.com/MODFLOW-USGS/modflow6/wiki/Parallel-MODFLOW-User-Guide



How to run it?

- Partitioning with Flo






```
sing base simulation
utils.Mf6Splitter(base_sim)
ng mask for the nr. of domains
t.optimize_splitting_mask(nparts=nr_domains)
del
it.split_model(split_array)
```

- Assign models to co
  - Better load balance
  - optional

```
i by Flopy

sim.hpc
```

sim.hpc:

```
# File generated by Flopy
BEGIN options
END options

BEGIN partitions
  model_0  0
  model_1  1
  model_2  2
  model_3  3
  model_4  4
  model_5  5
  model_6  6
  model_7  7
END partitions
```

- Run parallel simulation:

```
D:\> mpiexec.exe -np 8 mf6.exe -p
```

Notebook

step0_introduction.ipynb

Markdown | Notebook | Python 3 (ipykernel)

Introduction: running a watershed in parallel

These notebooks present the parallel capabilities recently added to MODFLOW 6. It is largely based on the hypothetical watershed presented in a recent paper on using FloPy to set up workflows for structured and unstructured MODFLOW models (https://doi.org/10.1111/gwat.13327). The following figure from that paper illustrates some of the properties and layout of the watershed: