Advanced ParFlow Short Course Overland Flow and Topographic Processing

October 3-4, 2019

University of Arizona

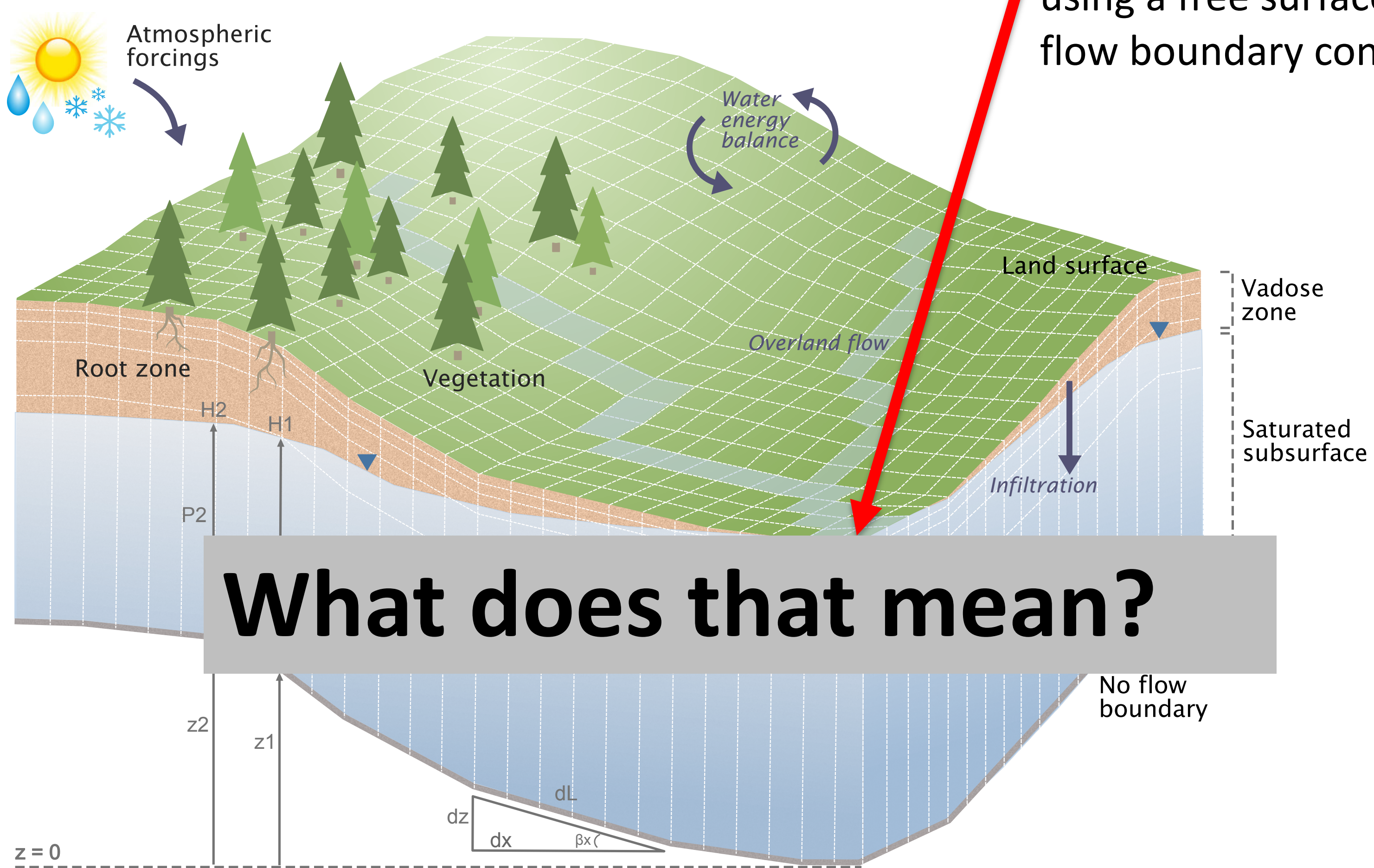
Laura Condon, University of Arizona
Nick Engdahl, Washington State University
Reed Maxwell, Colorado school of Mines

This material and short course was supported by the National Science Foundation and Department of Energy

Part 1: Overland Flow Boundary Conditions in ParFlow

ParFlow-CLM

Subsurface and overland flow are fully integrated in ParFlow using a free surface overland flow boundary condition



What does that mean?

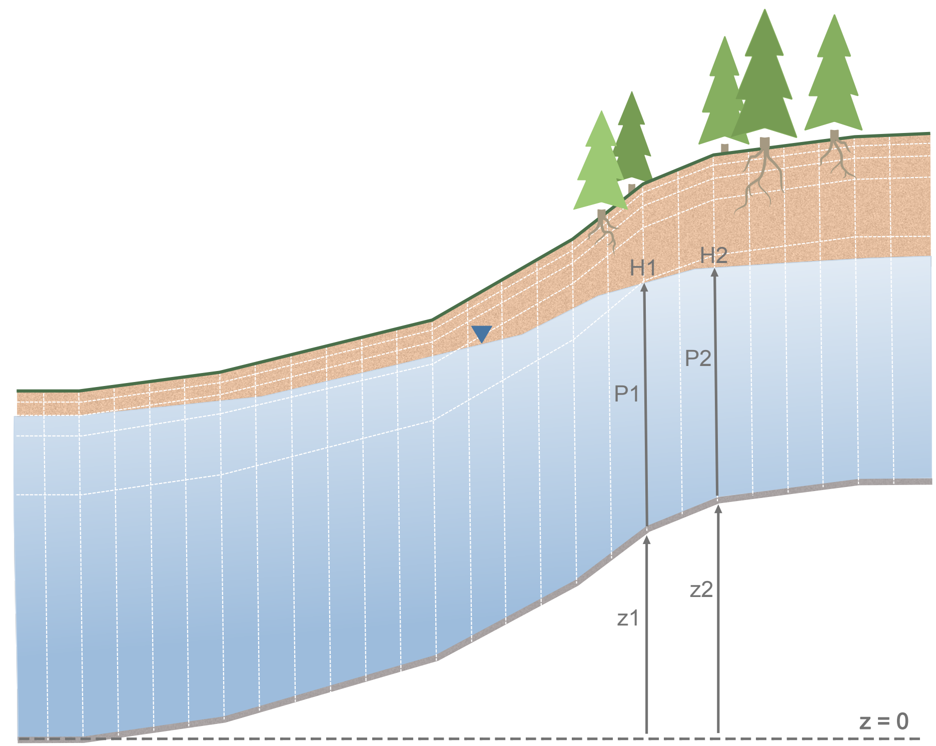
Maxwell (2013); Kollet and Maxwell (2008); Kollet and Maxwell (2006);Maxwell and Miller (2005); Dai et al. (2003); Jones and Woodward (2001); Ashby and Falgout (1996)

Free surface overland flow boundary condition

ParFlow solves Richards equation for 3D variably saturated flow everywhere in the subsurface

$$S_s S_w \frac{\partial \psi_p}{\partial t} + \phi \frac{\partial S_w(\psi_p)}{\partial t} = \nabla \cdot \mathrm{q} + q_s$$

$$\boldsymbol{q} = -k(x) k_r(\psi_p) \nabla(\psi_p - z)$$



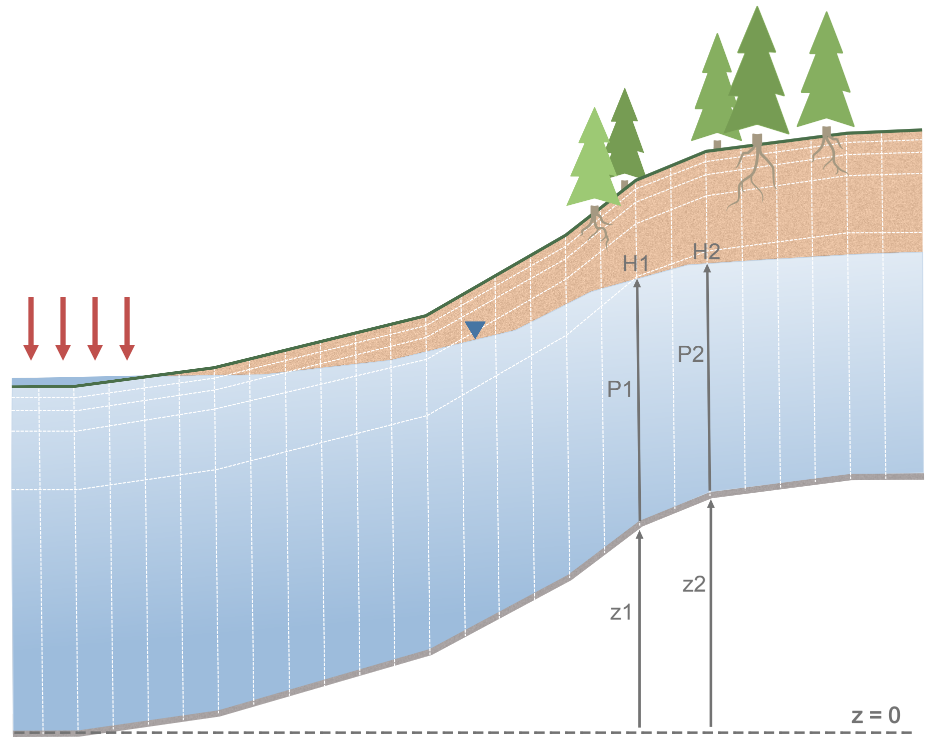
Φ = Prorosity, ψ_p= Pressure Head [L], S_w=water saturation, S_s=Specific storage [L^{-1}], k=Saturated hydraulic Conductivity [L/T], k_r=Relative Hydraulic Conductivity, q_s= source sink term [L^3/T], z=grid cell elevation [L], q=groundwater flux [L^2/T]

Free surface overland flow boundary condition

IFF there is ponded water in a cell in the top layer it swaps and solves overland flow for this grid cell instead using Manning's Equation

$$\frac{\partial \psi_s}{\partial t} = \nabla \vec{v} \psi_s + q_r$$

$$v_x = -\frac{\sqrt{S_{f,x}}}{n} \psi_s^{2/3}$$



ψ_s= Surface Ponding Depth [L], v=depth averaged velocity [L/T], n=Mannings roughness coefficient [T/L$^{1/3}$], S_f=friction slope

Free surface overland flow boundary condition

To swap back and forth at the boundary we:

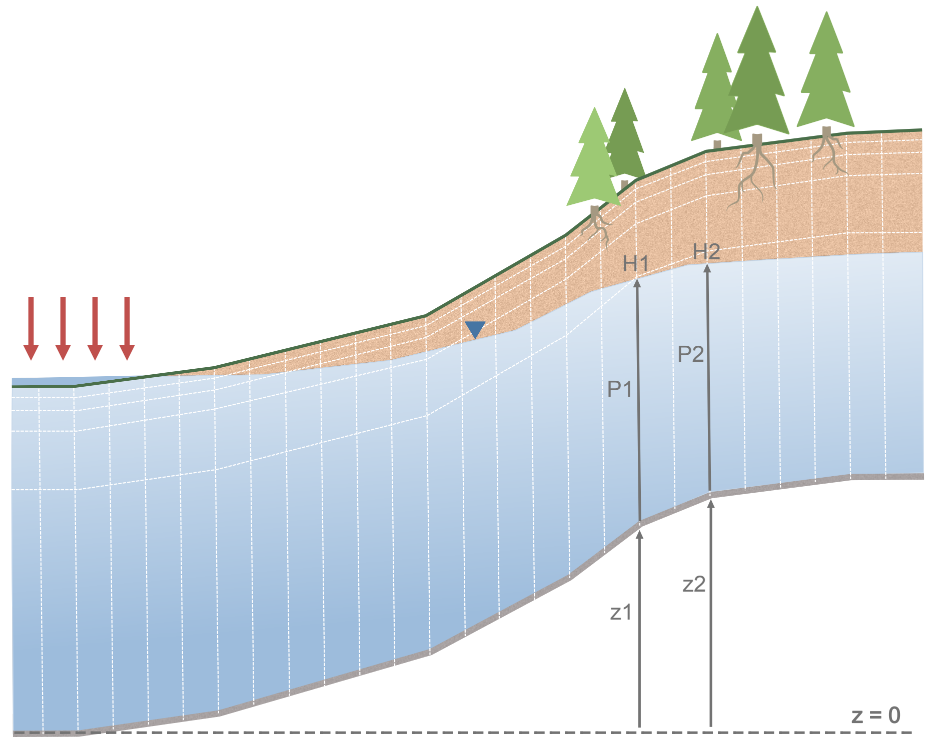
1) Assume continuity of pressure at the boundary between the surface an the subsurface: $\psi_p = \psi_s = \psi$

2) Add a flux for the subsurface exchange

$$\frac{\partial \psi_s}{\partial t} = \nabla \cdot (\vec{v}\psi_s) + q_r(x) + q_e(x)$$

$$\frac{\partial \parallel \psi, 0 \parallel}{\partial t} = \nabla \cdot (\vec{v} \parallel \psi, 0 \parallel) + q_r(x) + q_e(x)$$

ψ_s= Surface Ponding Depth [L], v=depth averaged velocity [L/T], n=Mannings roughness coefficient [T/$L^{1/3}$], S_f=friction slope, q_e=subsurface exchange flux



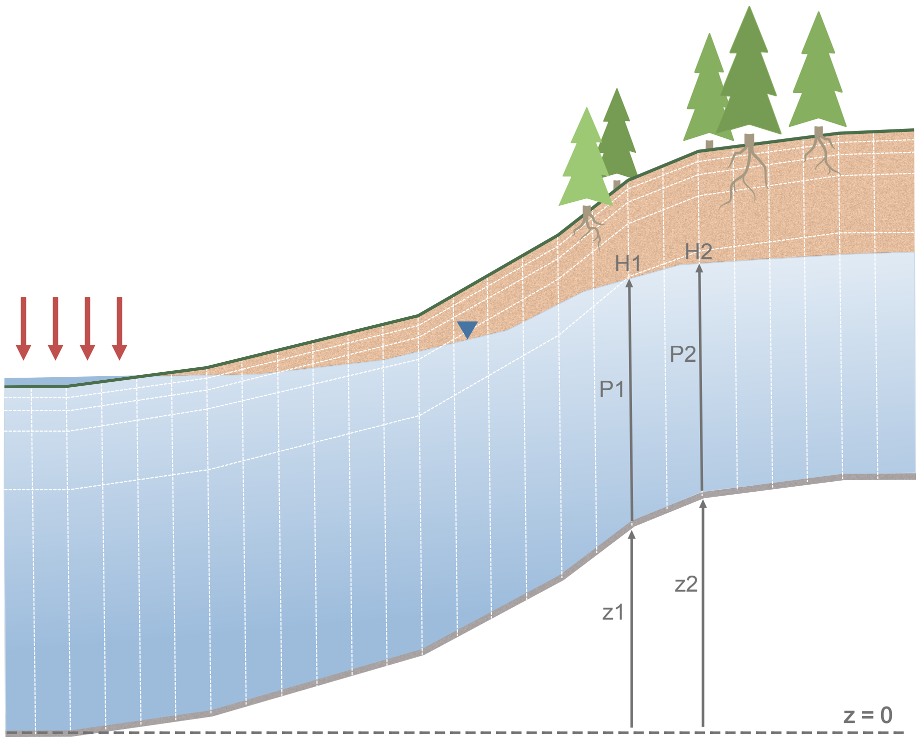
Refer to Kollet and Maxwell (2006) and Manual section 5.2-5.4 for complete Details

Kollet, S. J. and Maxwell, R. M. (2006). Integrated surface-groundwater flow modeling: A free-surface overland flow boundary condition in a parallel groundwater flow model. Advances in Water Resources, 29:945–958 .

What does this all mean for me?

Pros:

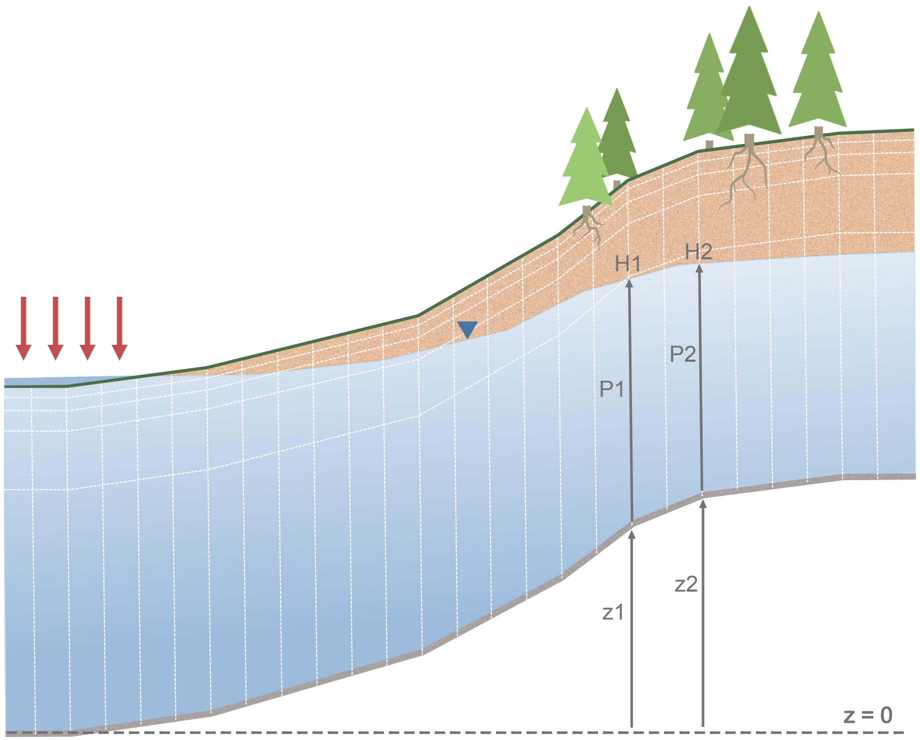
- This solution is fully integrated and fully mass conservative
- We do not specify the locations of surface water cells or set a surface water routing network a-priori
- Surface water bodies form within the simulation organically as as a result of groundwater convergence, saturation excess or infiltration excess overland flow
- In transient simulations surface water bodies can expand and shrink as a a function of water availability



What does this all mean for me?

Things to note:

- This means that the surface water network that forms in your domain will be very sensitive to the topography you set the model up with (i.e. your slopes)
- The solver has to do a lot of work for cells that are just on the edge of ponding where the solution is swapping between subsurface and overland flow
- Your streams can only be resolved at the resolution of your model (i.e. they will be at least the width of you model dx or dy)



Overland Flow Boundary Condition Options in ParFlow

1. OverlandFlow
2. OverlandKinematic
3. OverlandDiffusive
4. OverlandFlowPFB
5. SeepageFace

1. OverlandFlowPFB

- Set as options when you specify the boundary condition for the top surface of your domain
- Must be used in conjunction with Solver Richards
- Refer to Manual Section 6.1.25

Example:

pfset Patch.z-upper.BCPressure.Type OverlandFlow
pfset Solver Richards

Overland Flow Boundary Condition Options in ParFlow

1. OverlandFlow – Original overland flow boundary condition with kinematic wave formulation assuming cell centered slopes.
2. OverlandKinematic – Modified kinematic wave formulation assuming face centered slopes.
3. OverlandDiffusive – Diffusive wave formulation assuming face centered slopes consistent with OverlandKinematic
4. OverlandFlowPFB – Allows for gridded flux values to be read in.
5. SeepageFace – Allows flow to exit but lets keeps the surface pressure at zero. This is equivalent to turring on the spinup flag (pfset OverlandFlowSpinUp 1)

Kinematic and diffusive wave solutions

- OverlandFlow, OverlandKinematic and OverlandDiffusive
- All solving Manning's equation:

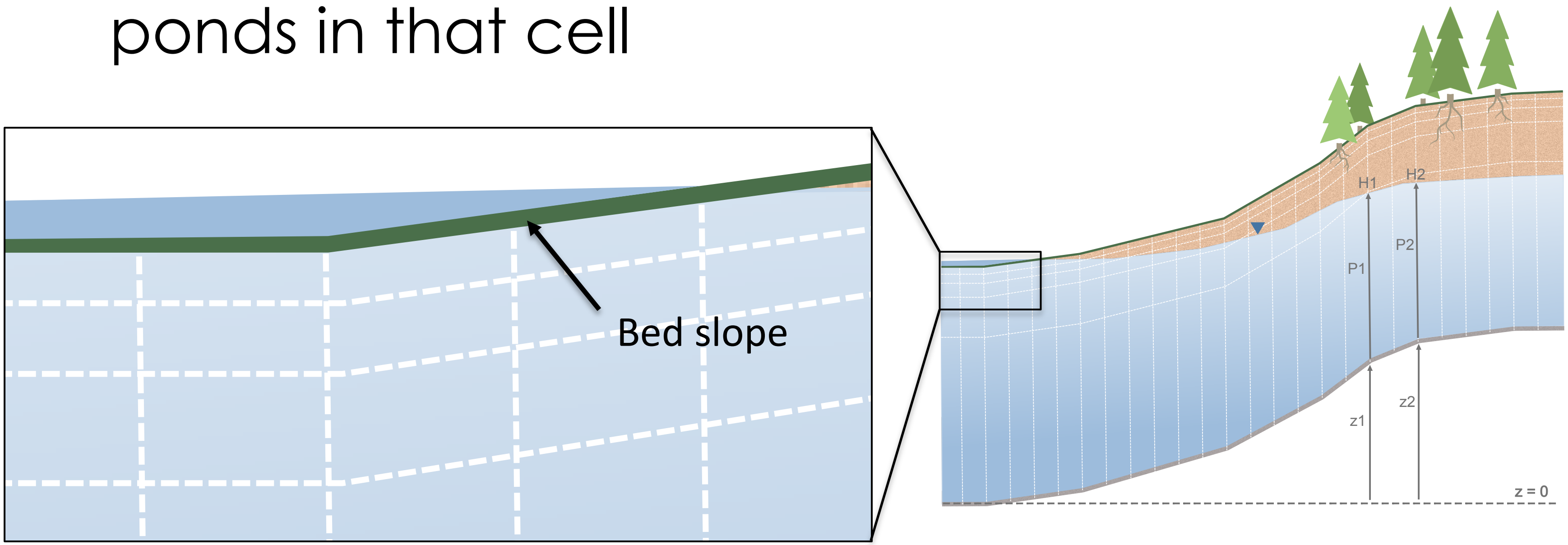
$$\boldsymbol{v}_x = -\frac{\sqrt{S_{f,x}}}{n}\psi_s^{2/3}$$

- The difference is in the formulation of the friction slope and the assumptions made about how the slopes you give to ParFlow are calculated

OverlandFlow and OverlandKinematic both use the kinematic wave approximation

Friction Slope (S_f) = Bed Slope (S_o)

- S_o is the topographic slope you give to ParFlow
- This means if you have a flat cell water will not flow out of it regardless of how much water ponds in that cell



Two key differences between OverlandFlow and OverlandKinematic:

1. Formulation of Manning's Equation

OverlandFlow	$v_x = -\frac{\sqrt{S_{f,x}}}{n}\psi_s^{2/3}$
OverlandKinematic	$v_x = -\frac{S_{f,x}}{n\sqrt{\bar{S}_f}}\psi_s^{2/3}$

$\bar{S}_f$ = Magnitude of S_f

- These two formulations very similar but the OverlandKinematic has the magnitude of the 2D slopes in the denominator and can provide better numerical stability
- *Because the slope is in the denominator here we have to add an epsilon if the slope goes to zero:

double **Solver.OverlandKinematic.Epsilon** [1E-5]

This key provides a minimum value for the $\bar{S}_f$ used in the **OverlandKinematic** boundary condition.

Example Useage:

```
pfset Solver.OverlandKinematic.Epsilon 1E-7
```

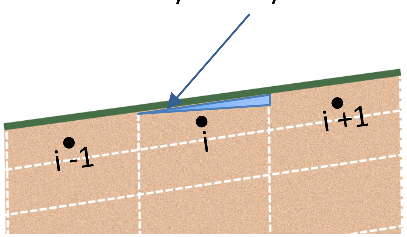
Two key differences between OverlandFlow and OverlandKinematic:

2. Assumptions about slope calculations

OverlandFlow

Assumes cell centered slopes calculated from face elevations

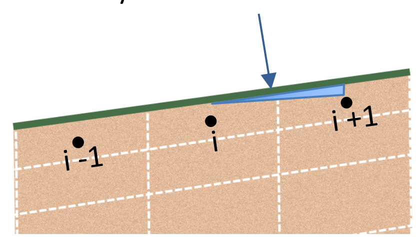
$So_i=(z_{i+1/2}-z_{i-1/2})/dx$



OverlandFlow

Assumes face centered slopes calculated grid centered elvations

$So_{i+1/2}=(z_{i+1}-z_i)/dx$



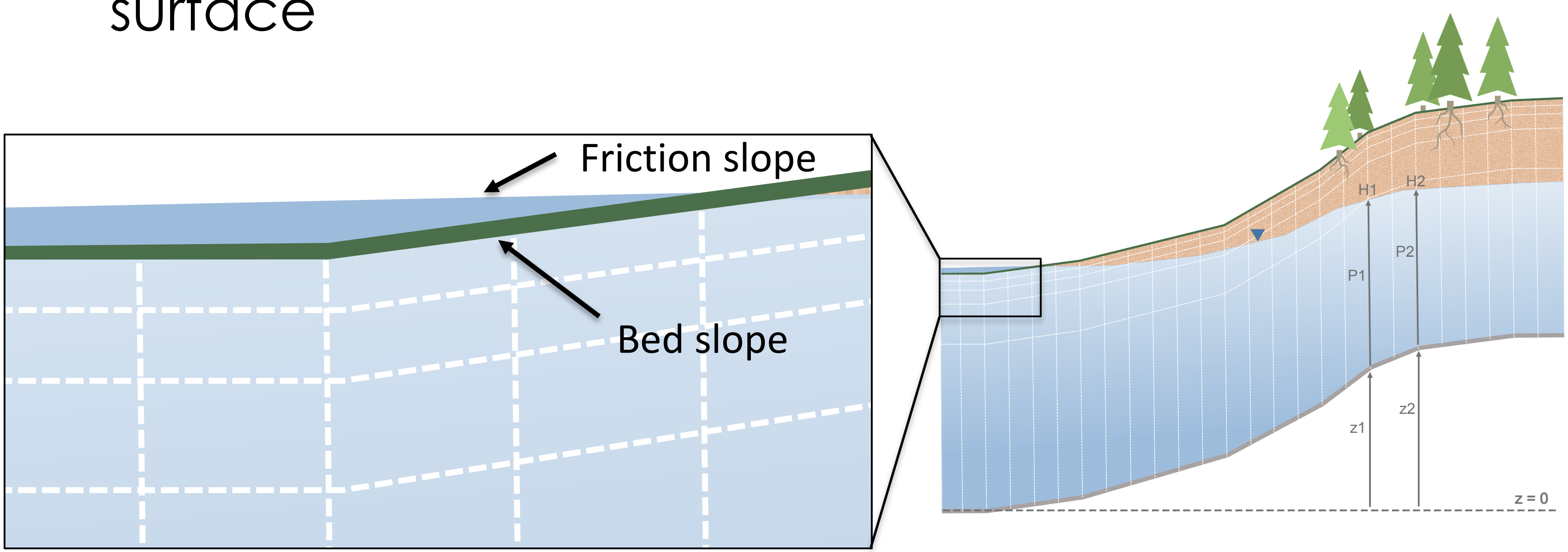
If the slopes are processed correctly these two approaches should yield very similar solutions.

- OverlandKinematic provides better consistency between the surface and subsurface
- If you have cell centered elevations for your DEM calculating slopes as i+1-i results in cell centered slopes so this can be a more intuitive way to formulate the slopes

OverlandDiffusive uses the friction slope instead of the bed slope

Friction Slope (S_f) = Slope of ponded surface (i.e. bed slope + ponded depth)

- This means if you have a flat cell water will flow out based on the gradient of the ponded water surface



OverlandDiffusive uses similar formulation to OverlandKinematic

Same Manning's formulation

OverlandDiffusive $$v_x = -\frac{S_{f,x}}{n\sqrt{\bar{S_f}}}\psi_s^{2/3}$$

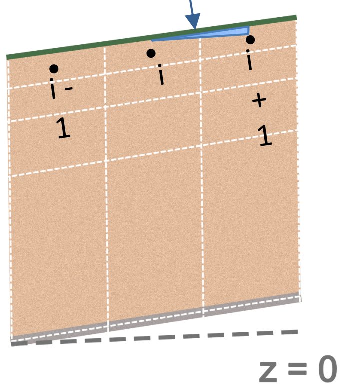
Similar to OverlandKinematic we also have an epsilon if the slope goes to zero:

double **Solver.OverlandDiffusive.Epsilon** [1E-5]
This key provides a minimum value for the $\bar{S_f}$ used in the **OverlandDiffusive** boundary condition.
Example Useage:

```
pfset Solver.OverlandDiffusive.Epsilon 1E-7
```

Bed slopes are assumed to be face centered the same as OverlandKinematic

$So_{i+1/2}=(z_{i+1}-z_i)/dx$



Overland Flow Exercises

Part 2: Topographic Processing

Goals:

1. Get a fully connected drainage network
2. Smooth the DEM to adjust for noise caused by resolution
3. Calculate slopes (i.e. the actual input needed by ParFlow)

This is a well known problem and there are many approaches to address this

Contents lists available at ScienceDirect
Computers & Geosciences
journal homepage: www.elsevier.com/locate/cageo
ELSEVIER

Routing overland flow through sinks and flats in interpolated raster terrain surfaces ☆

Frank Kenny *, Bryce Matthews, Kent Todd
Water Resources Information Program, Ontario Ministry of Natural Resources, 300 Water St., Peterborough, Ontario, Canada K9J 8M5

HYDROLOGICAL PROCESSES
Hydrol. Process. **30**, 846–857 (2016)
Published online 20 September 2015 in Wiley Online Library
(wileyonlinelibrary.com) DOI: 10.1002/hyp.10648

Efficient hybrid breaching-filling sink removal methods for flow path enforcement in digital elevation models

John B. Lindsay*
Department of Geography, The University of Guelph, 50 Stone Road East, Guelph N1G 2W1, Canada

EARTH SURFACE PROCESSES AND LANDFORMS
Earth Surf. Process. Landforms **41**, 658–668 (2016)
Copyright © 2015 John Wiley & Sons, Ltd.
Published online 13 January 2016 in Wiley Online Library
(wileyonlinelibrary.com) DOI: 10.1002/esp.3888

The practice of DEM stream burning revisited

John B. Lindsay*
Department of Geography, University of Guelph, Guelph, Canada

Received 27 July 2015; Revised 7 December 2015; Accepted 7 December 2015

*Corre

Contents lists available at ScienceDirect
Computers & Geosciences
journal homepage: www.elsevier.com/locate/cageo
ELSEVIER

Priority-flood: An optimal depression-filling and watershed-labeling algorithm for digital elevation models

CrossMark

Richard Barnes [a,*], Clarence Lehman [b], David Mulla [c]
[a] *Ecology, Evolution, & Behavior, University of Minnesota, USA*
[b] *College of Biological Sciences, University of Minnesota, USA*
[c] *Soil, Wat*

Hydrol. Earth Syst. Sci., 14, 1153–1165, 2010
www.hydrol-earth-syst-sci.net/14/1153/2010/
doi:10.5194/hess-14-1153-2010

Hydrology and Earth System Sciences

On the uncertainty of stream networks derived from elevation data: the error propagation approach

T. Hengl[1], G. B. M. Heuvelink[2], and E. E. van Loon[1]

WATER RESOURCES RESEARCH, VOL. 34, NO. 4, PAGES 897–901, APRIL 1998

Extraction and representation of nested catchment areas from digital elevation models in lake-dominated topography

D. Scott Mackay
Department of Forest Ecology and Management, and Institute for Environmental Studies
University of Wisconsin–Madison

Lawrence E. Band[1]
Department of Geography, University of Toronto, Toronto, Ontario, Canada

This is a well known problem and there are many approaches to address this

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

HYDROLOGICAL PROCESSES
Hydrol. Process. **30**, 846–857 (2016)
Published online 20 September 2015 in Wiley Online Library

Pit filling algorithms ensure drainage and can achieve globally optimal solutions but may not match the drainage network

The practice of DEM stream burning revisited

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

Stream burning methods match the spatial configuration of the drainage network but can result in deeply incised channels that alter the terrain attributes of the DEM

, APRIL 1998

On the uncertainty of stream networks derived from elevation data: the error propagation approach

T. Hengl[1], G. B. M. Heuvelink[2], and E. E. van Loon[1]

Extraction and representation of nested catchment areas from digital elevation models in lake-dominated topography

D. Scott Mackay
Department of Forest Ecology and Management, and Institute for Environmental Studies
University of Wisconsin–Madison

Lawrence E. Band[1]
Department of Geography, University of Toronto, Toronto, Ontario, Canada

Processing Topography: PriorityFlood Tool

GitHub Repo

https://github.com/lecondon/PriorityFlow

README.md

PriorityFlow

PriorityFlow is a toolkit for topographic processing for hydrologic models. This repo contains a set of R Functions and example workflow scripts.

Development Team

- Laura Condon (lecondon@email.arizona.edu)
- Reed Maxwell (rmaxwell@mines.edu)

Citation

For more details on the model and if you use PriorityFlow in published work please cite the following reference:
Condon, LE and RM Maxwell (2019). Modified priority flood and global slope enforcement algorithm for topographic processing in physically based hydrologic modeling applications. Computers & Geosciences,
doi

DE

The
(Wa
eve

As i
pro

ELSEVIER
Computers & Geosciences
Volume 126, May 2019, Pages 73-83

Modified priority flood and global slope enforcement algorithm for topographic processing in physically based hydrologic modeling applications ☆

Laura E. Condon [a], Reed M. Maxwell [b]

- Modified priority flood algorithm for depression filling and stream enforcement
- Slope processing tools and workflows for D4 and D8 routing
- Options for stream network smoothing
- Additional options for handling primary and secondary flow directions
- More control of DEM Processing and generated PF Consistent slope outputs

Step 1: Priority Flood Algorithm

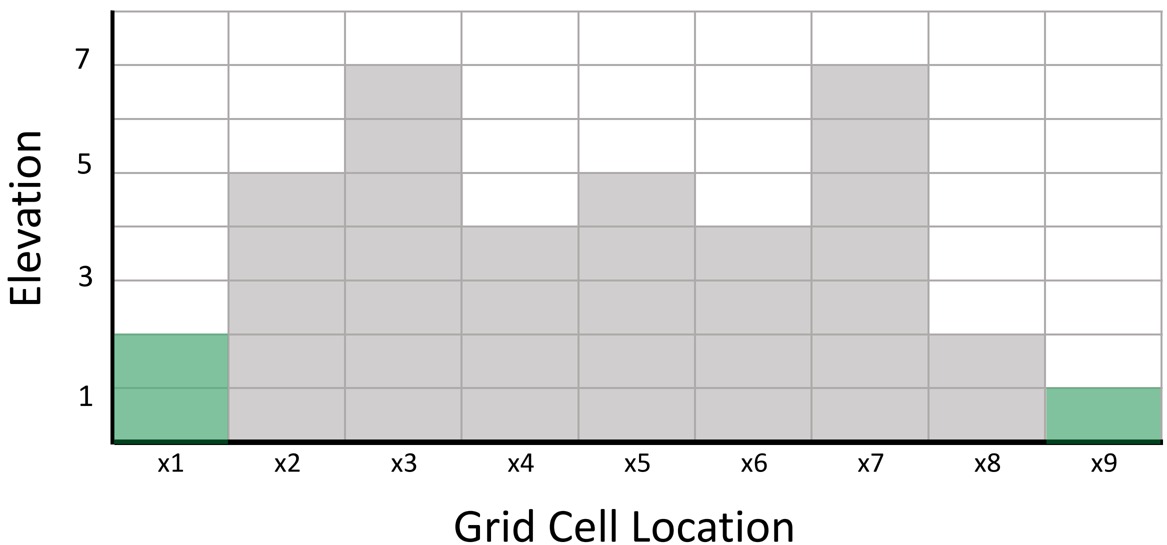
Adjust the DEM to ensure a fully connected D4 drainage network

Outputs:

- Adjusted DEM with all pits filled
- Primary flow directions

Priority Flood Algorithm

1. Initialize a 'Priority Queue' with all of the grid cells on the boundary of the domain



Priority Queue:

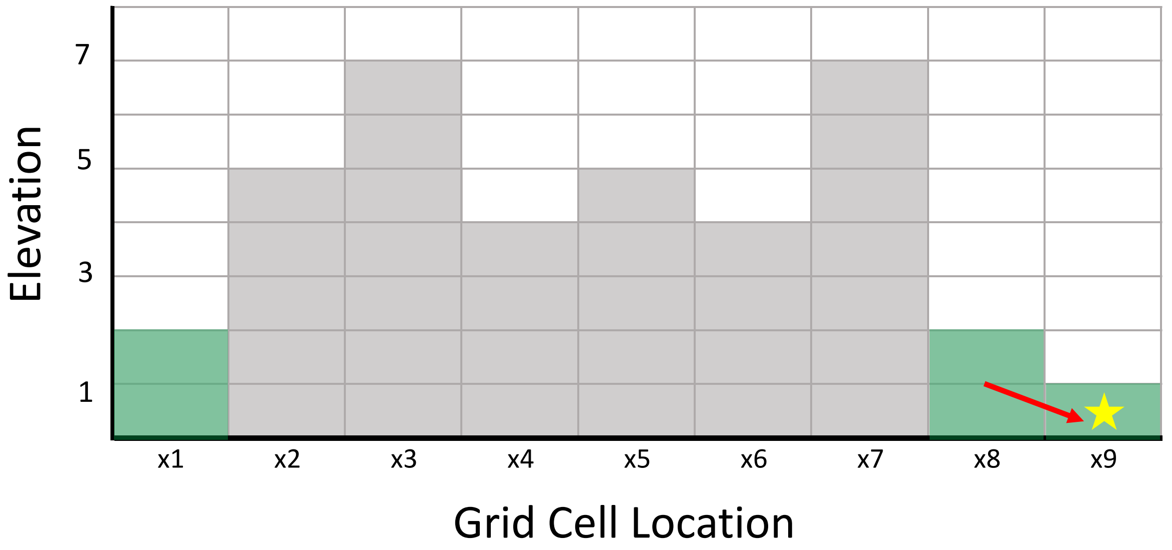
X9 = 1

X1 = 2

Priority Flood Algorithm

2. Process the lowest elevation cell on the priority queue

Add any unprocessed neighbors to the queue increasing the elevation if needed to ensure it can drain to the current cell



Priority Queue:

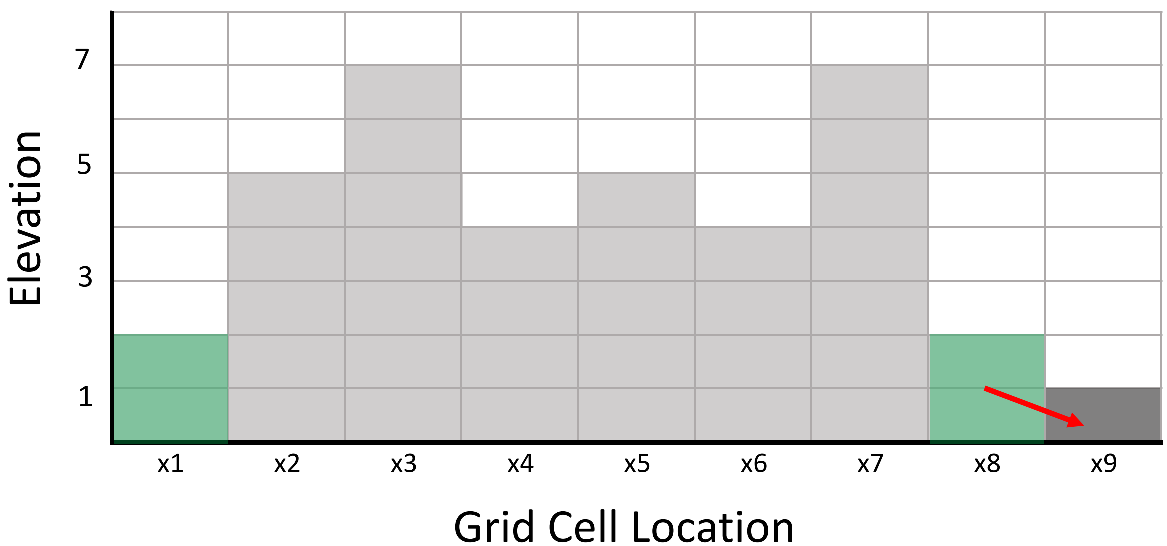
X9 = 1

X1 = 2

X8 = 2

Priority Flood Algorithm

3. Mark the current cell as processed and remove from the priority queue



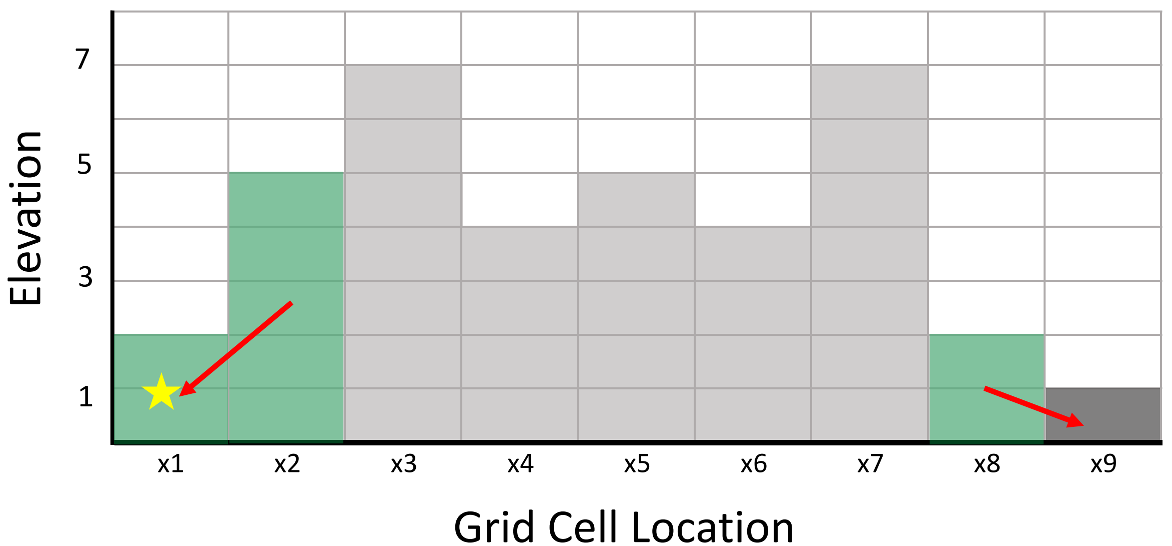
Priority Queue:

X1 = 2

X8 = 2

Priority Flood Algorithm

4. Repeat steps 2 and 3 until all of the cells have been processed



Priority Queue:

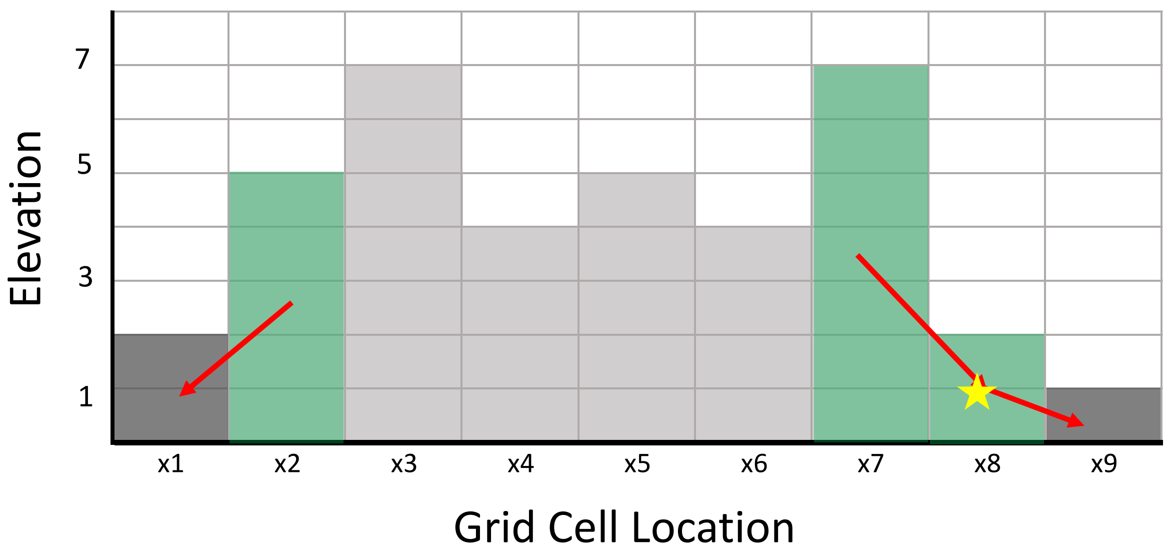
X1 = 2

X8 = 2

X2 = 5

Priority Flood Algorithm

4. Repeat steps 2 and 3 until all of the cells have been processed

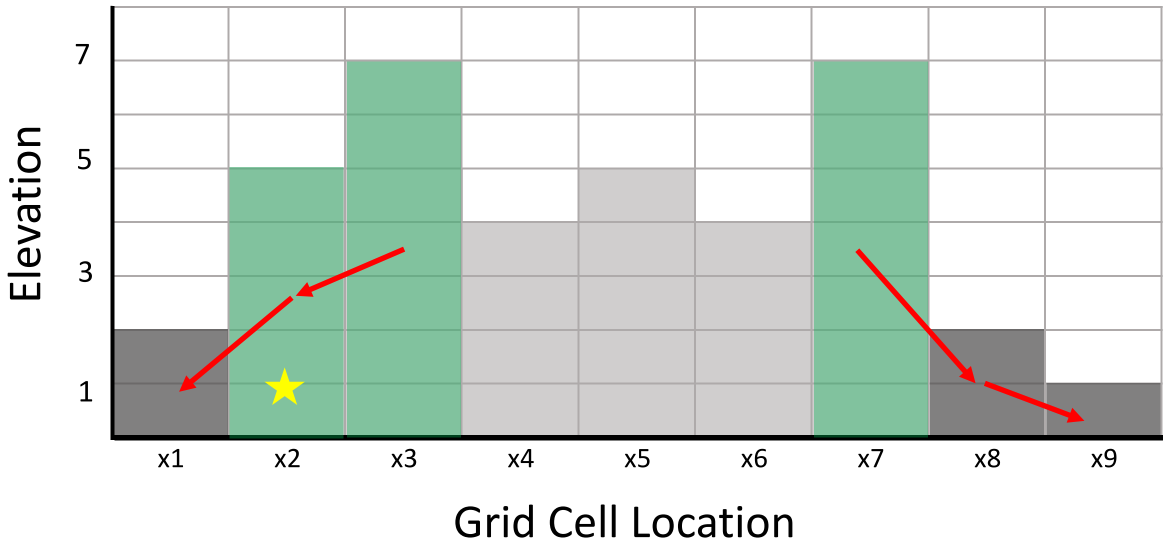


Priority Queue:

X8 = 2
X2 = 5
X7 = 7

Priority Flood Algorithm

4. Repeat steps 2 and 3 until all of the cells have been processed

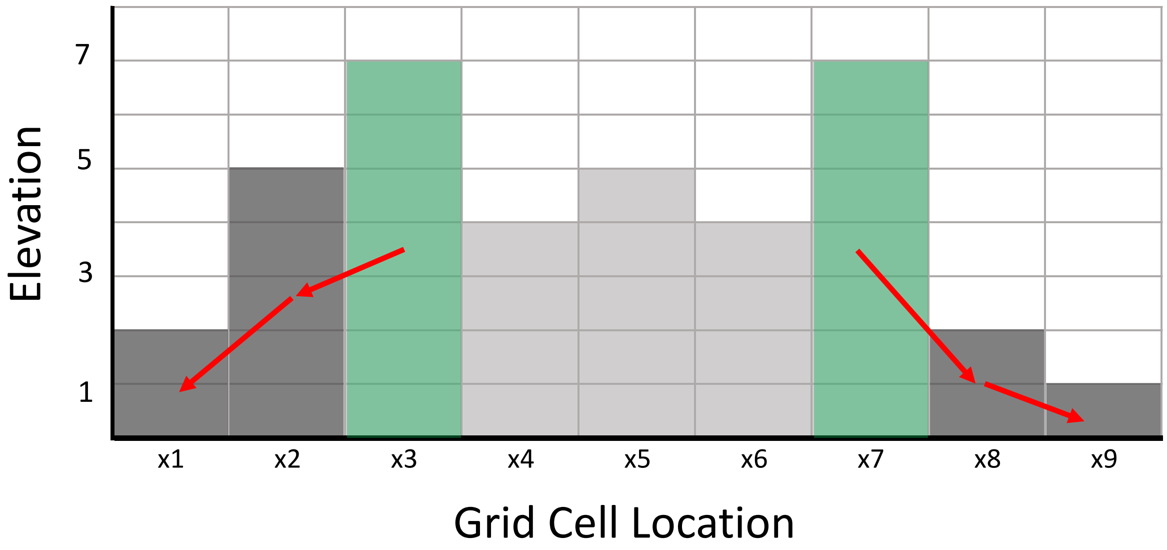


Priority Queue:

X2 = 5
X7 = 7
X3 = 7

Priority Flood Algorithm

4. Repeat steps 2 and 3 until all of the cells have been processed

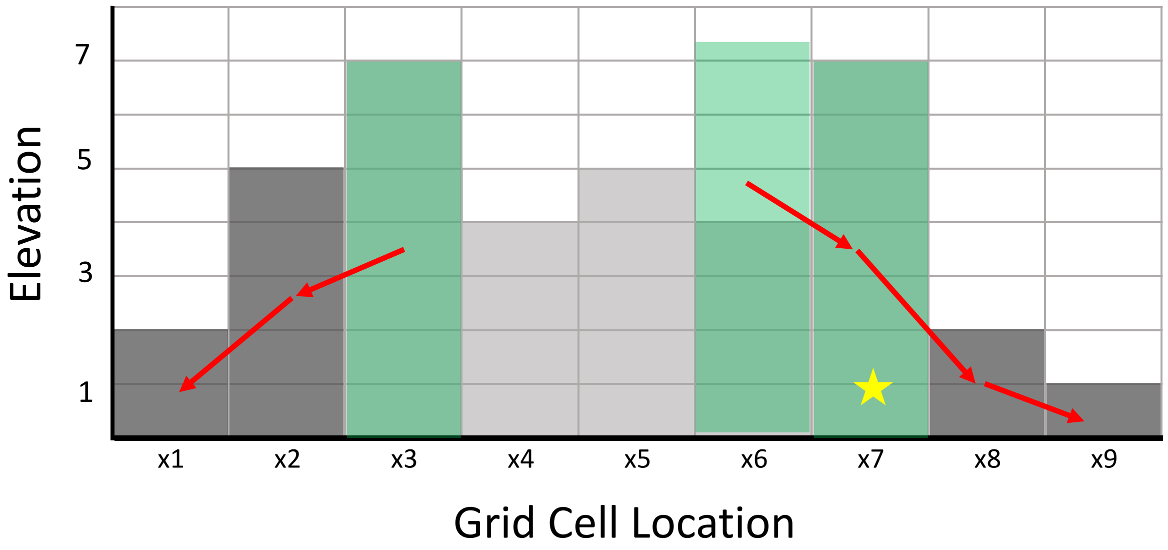


<u>Priority Queue:</u>

X7 = 7
X3 = 7

Priority Flood Algorithm

4. Repeat steps 2 and 3 until all of the cells have been processed

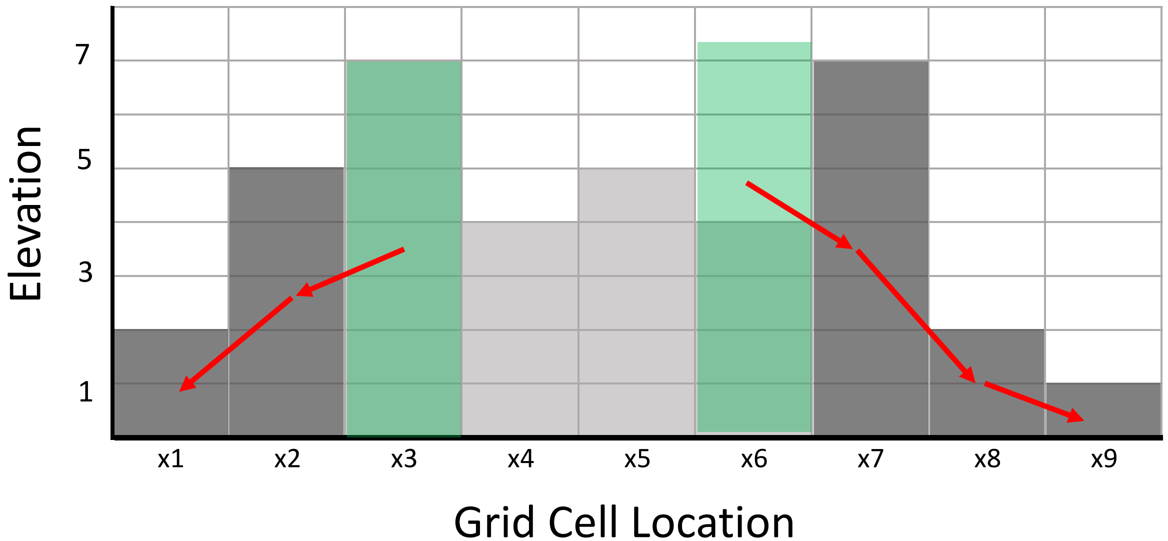


Priority Queue:

X7 = 7
X3 = 7
X6 = 7.1

Priority Flood Algorithm

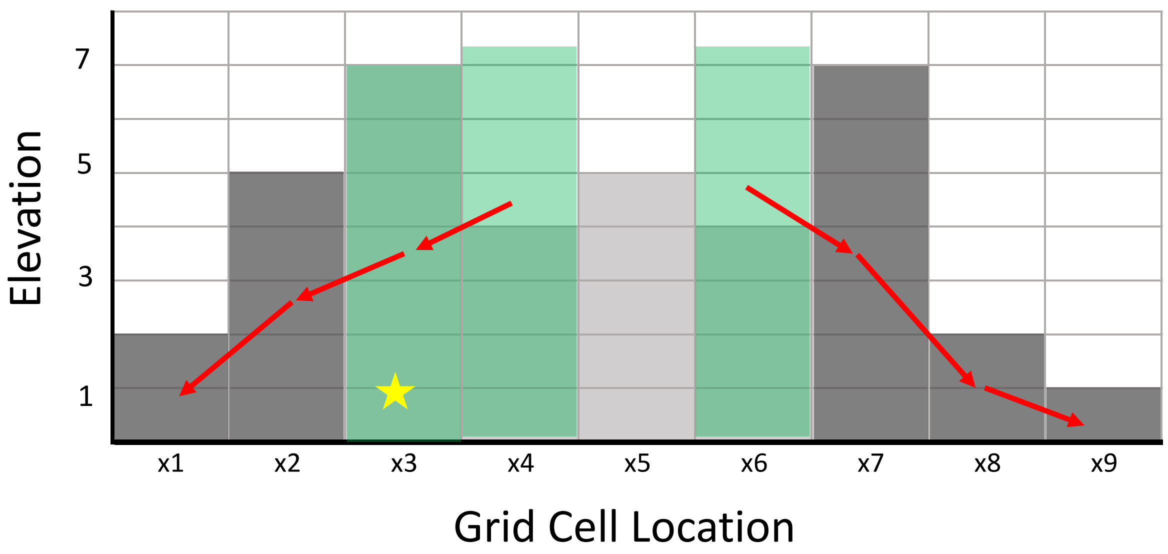
4. Repeat steps 2 and 3 until all of the cells have been processed



Priority Queue:
X3 = 7
X6 = 7.1

Priority Flood Algorithm

4. Repeat steps 2 and 3 until all of the cells have been processed



Priority Queue:

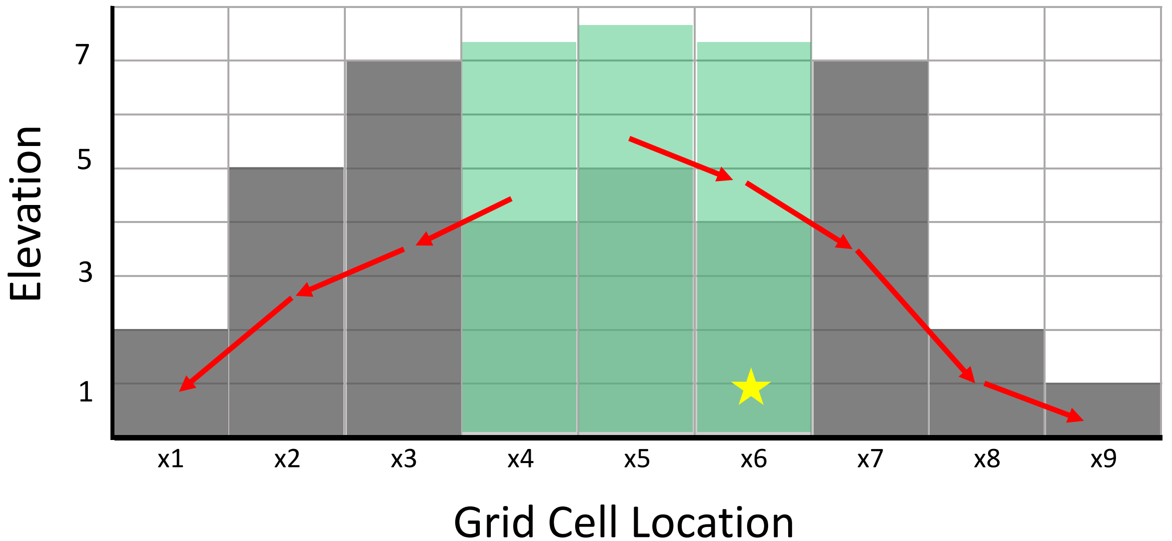
X3 = 7

X6 = 7.1

X4 = 7.1

Priority Flood Algorithm

4. Repeat steps 2 and 3 until all of the cells have been processed



Priority Queue:

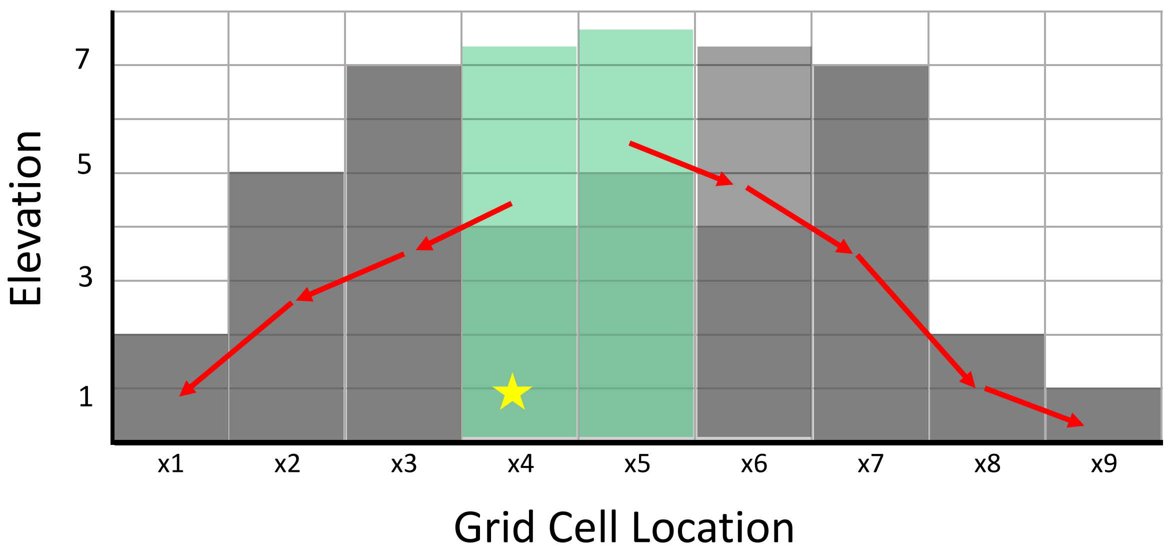
X6 = 7.1

X4 = 7.1

X5 = 7.2

Priority Flood Algorithm

4. Repeat steps 2 and 3 until all of the cells have been processed



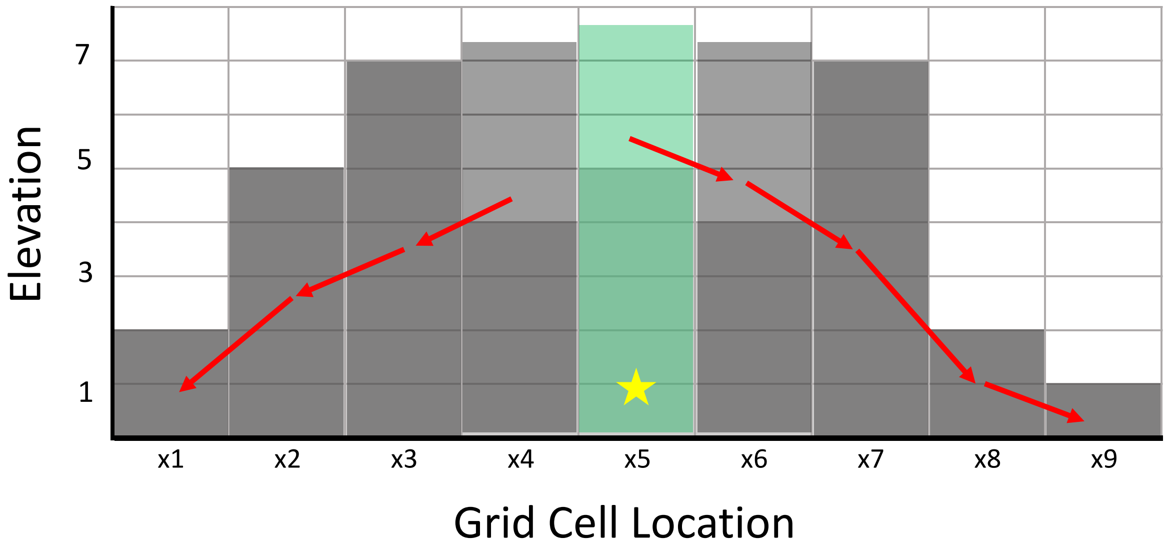
Priority Queue:

X4 = 7.1

X5 = 7.2

Priority Flood Algorithm

4. Repeat steps 2 and 3 until all of the cells have been processed

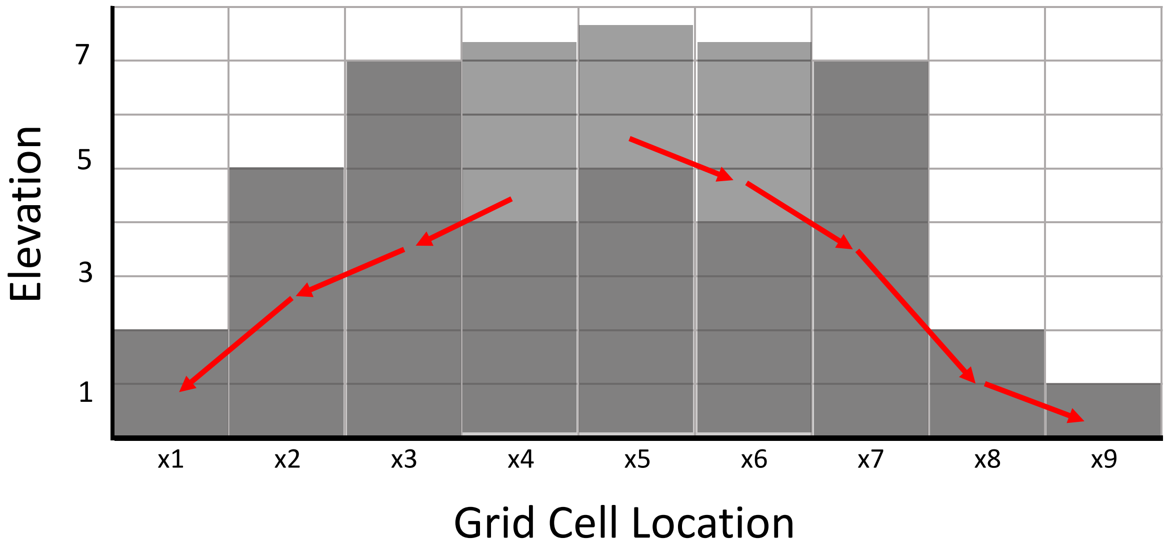


Priority Queue:

X5 = 7.2

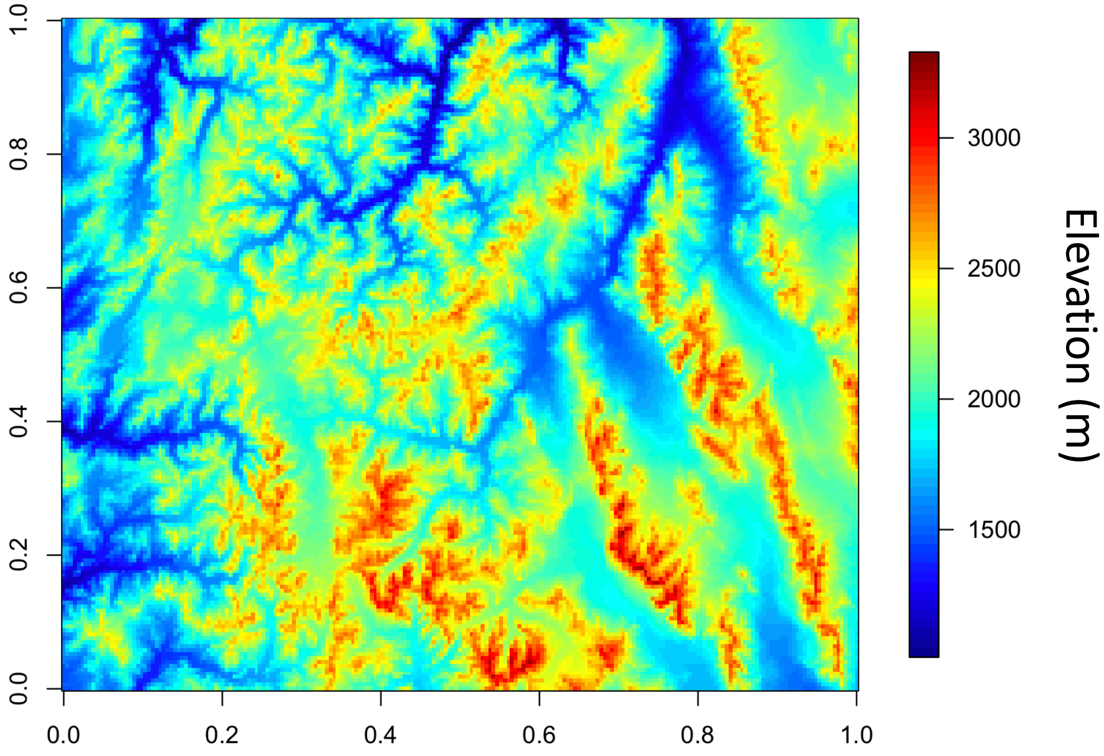
Priority Flood Algorithm

4. Repeat steps 2 and 3 until all of the cells have been processed



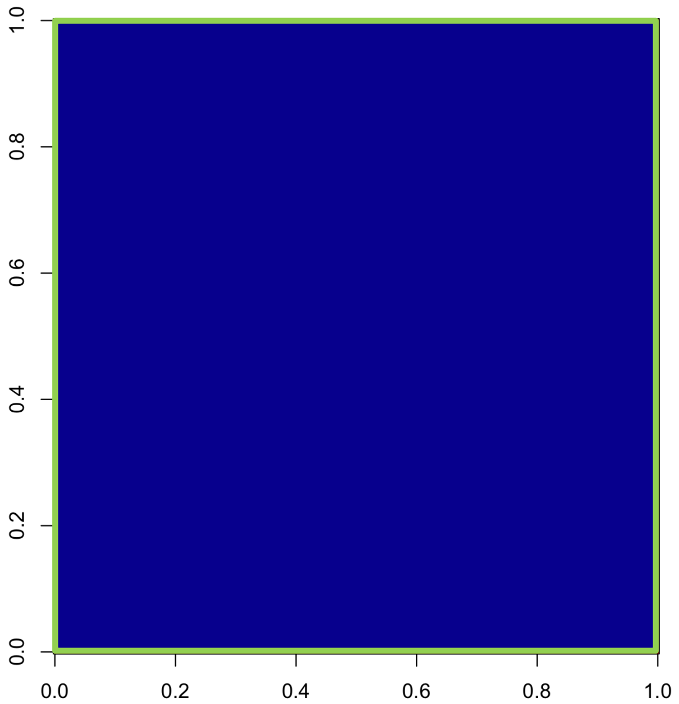
Priority Queue:

Priority Flood Algorithm in 2D

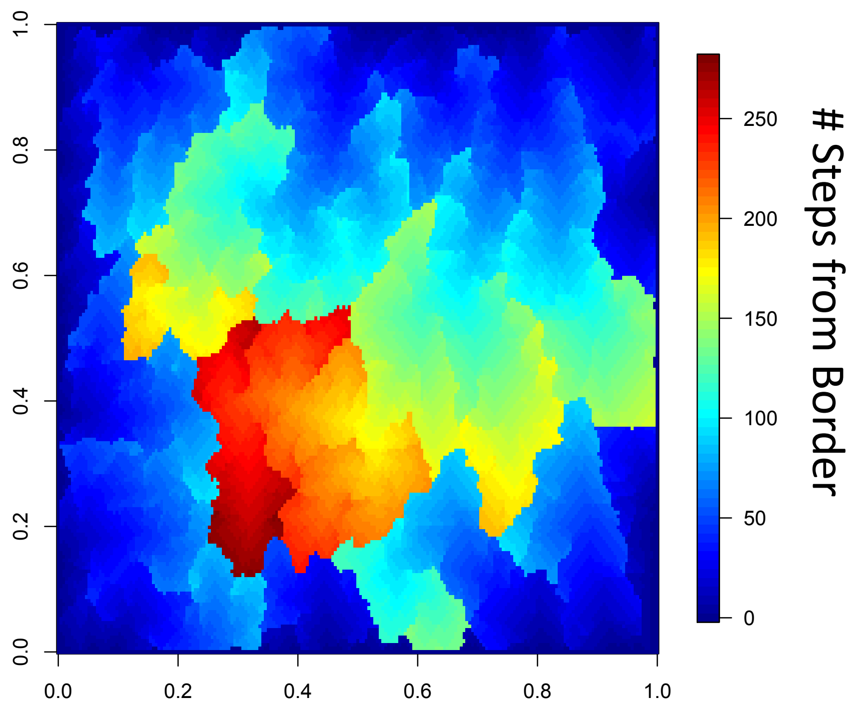


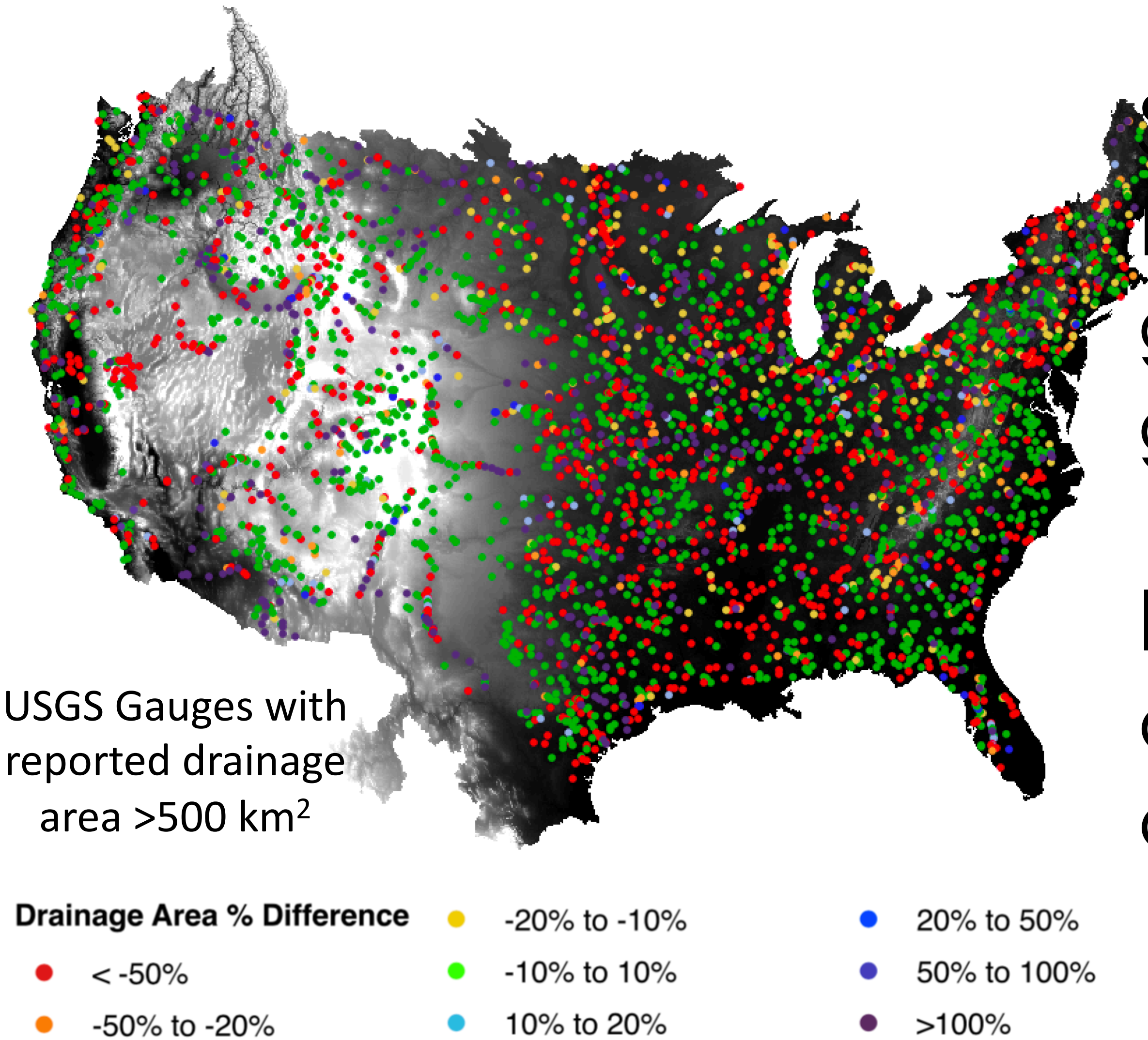
2D Priority Flood

1. Initialize queue with all border cells



2. Work upstream as with 1D example but adding D4 neighbors

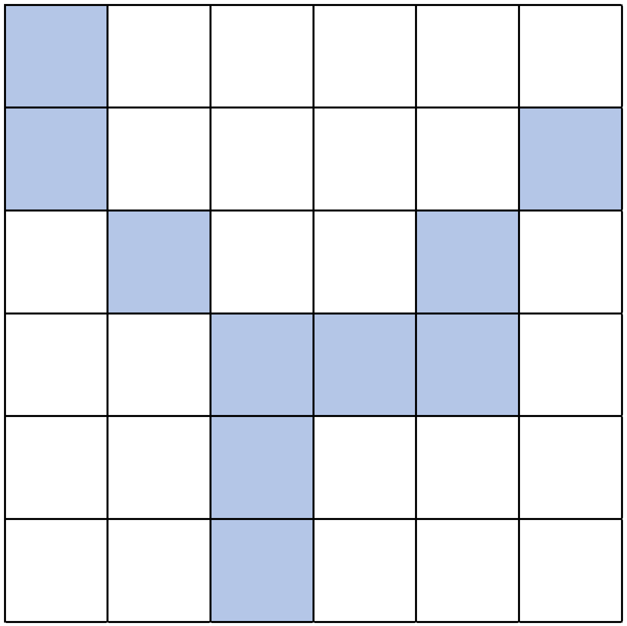




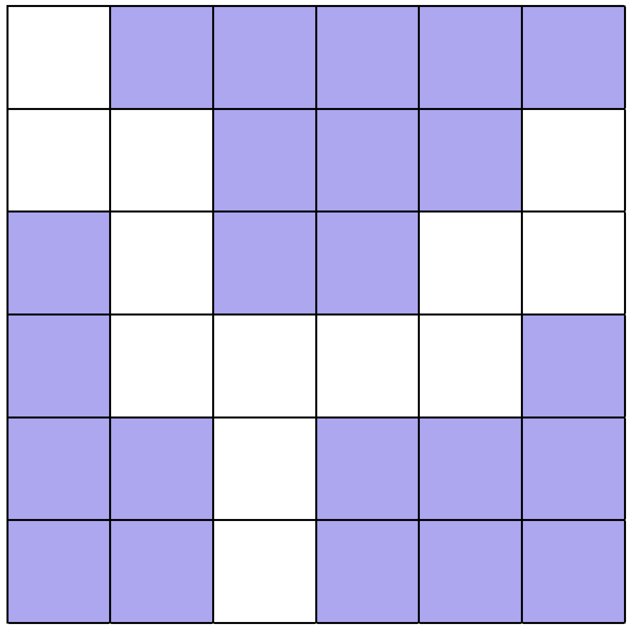
Standard Priority Flood processing guarantees globally 'optimal' solution but may not optimally capture drainage networks

Modified priority flood to improve stream network representation

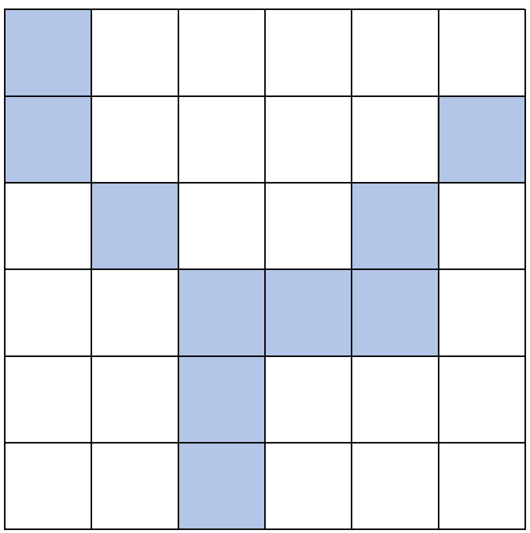
Create a mask of stream cells and process only the cells on this mask first

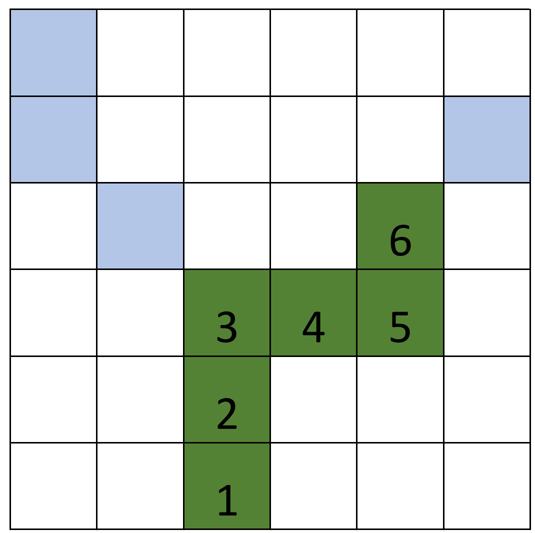


Process the remaining cells treating the stream networks as the boundary

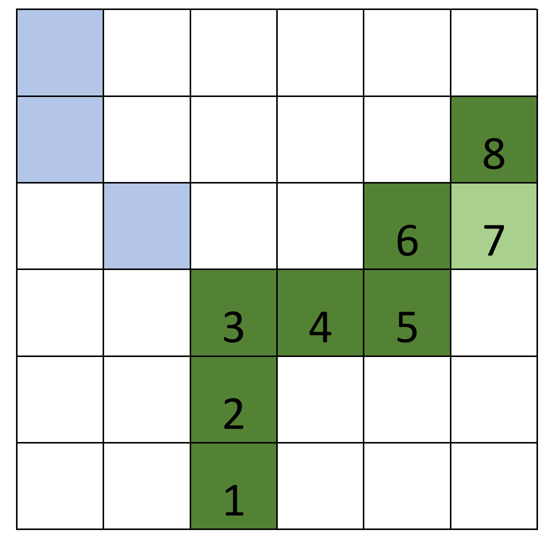


Processing for a D4 Connected drainage network



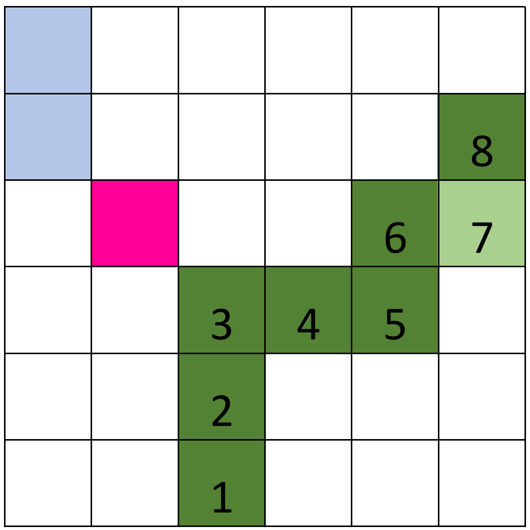


Walk upstream adding D4 neighbors on the stream mask to the queue and processing

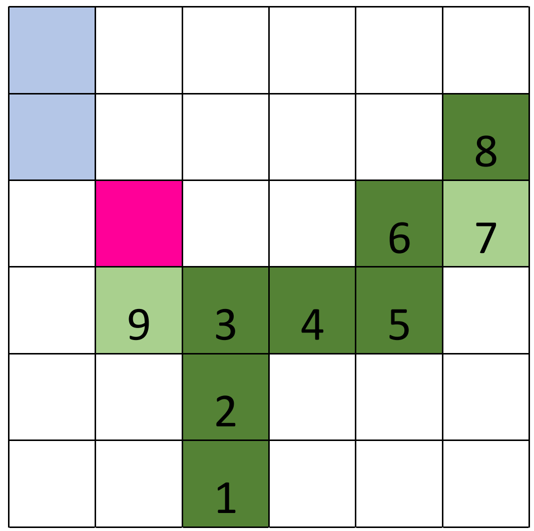


If only D8 neighbor exists add the lowest cost D4 path to this neighbor to the stream mask

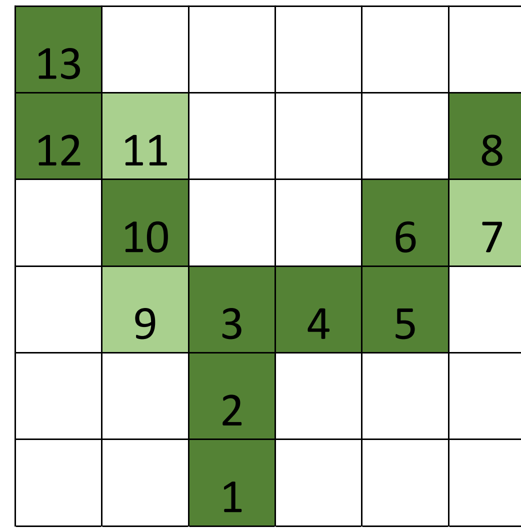
Processing for a D4 Connected drainage network

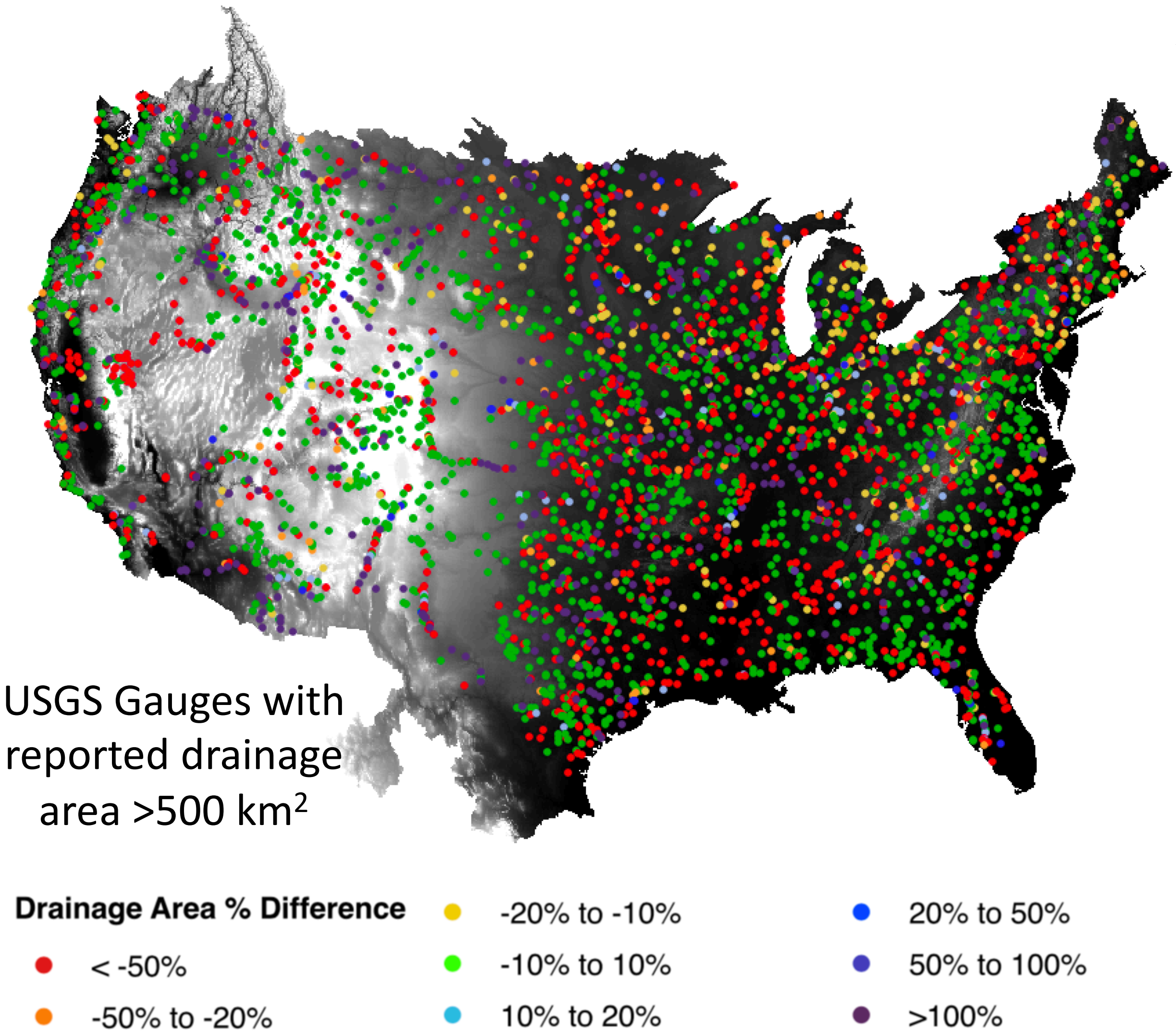


Check for any orphaned branches

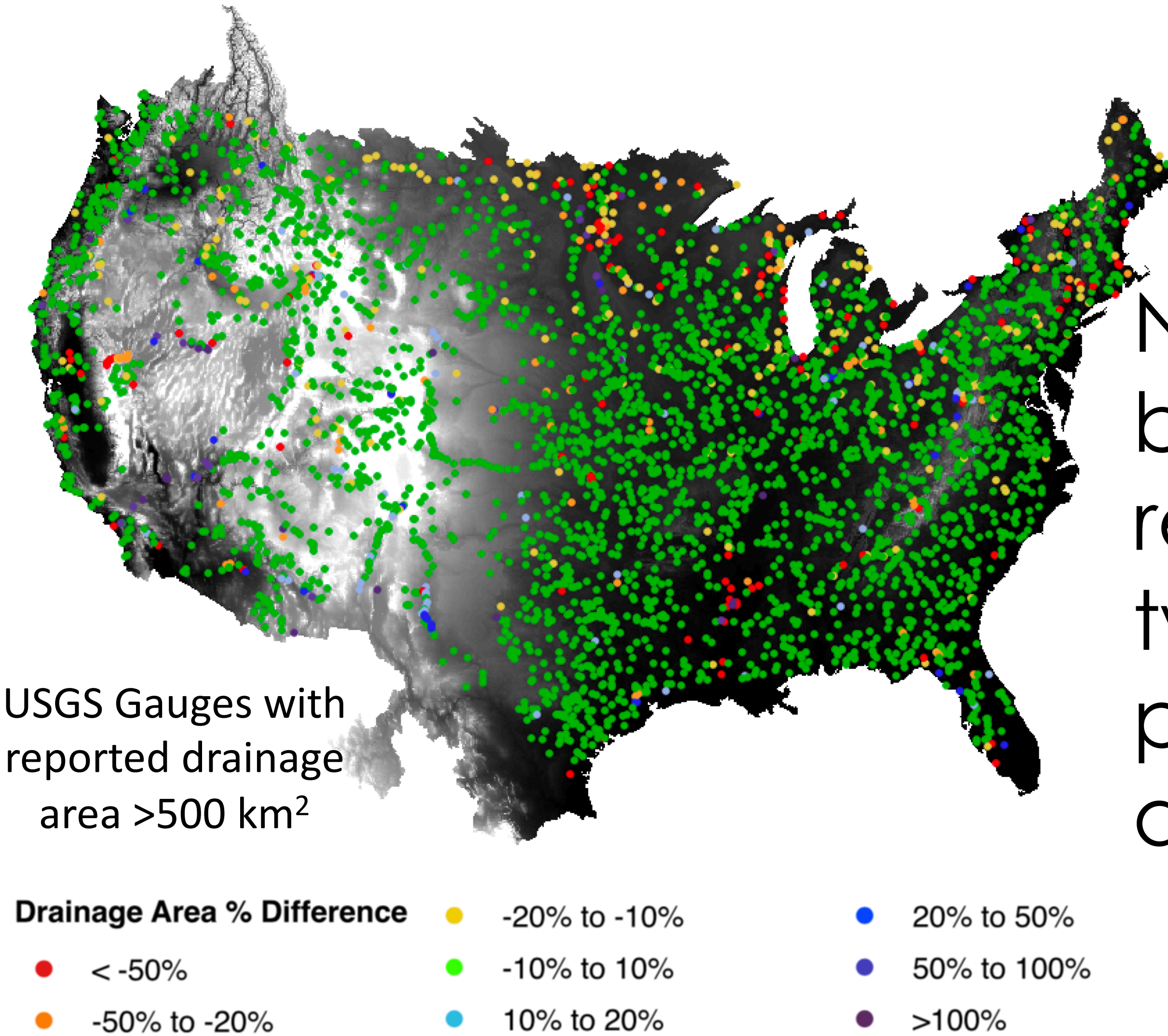


Add the lowest cost D4 path to thisbranch to the stream mask



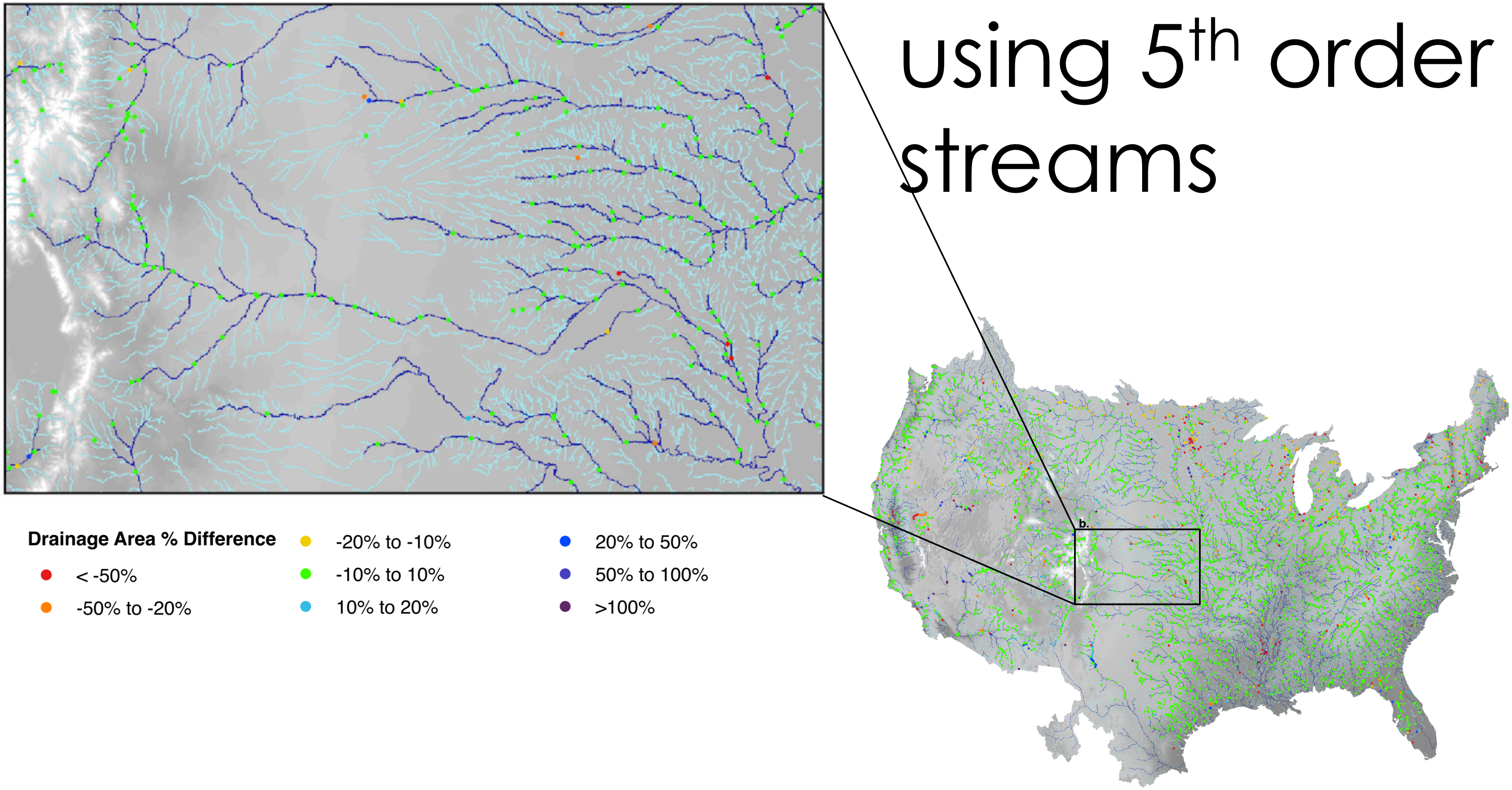


Standard Priority Flood processing guarantees globally 'optimal' solution but may not optimally capture drainage networks



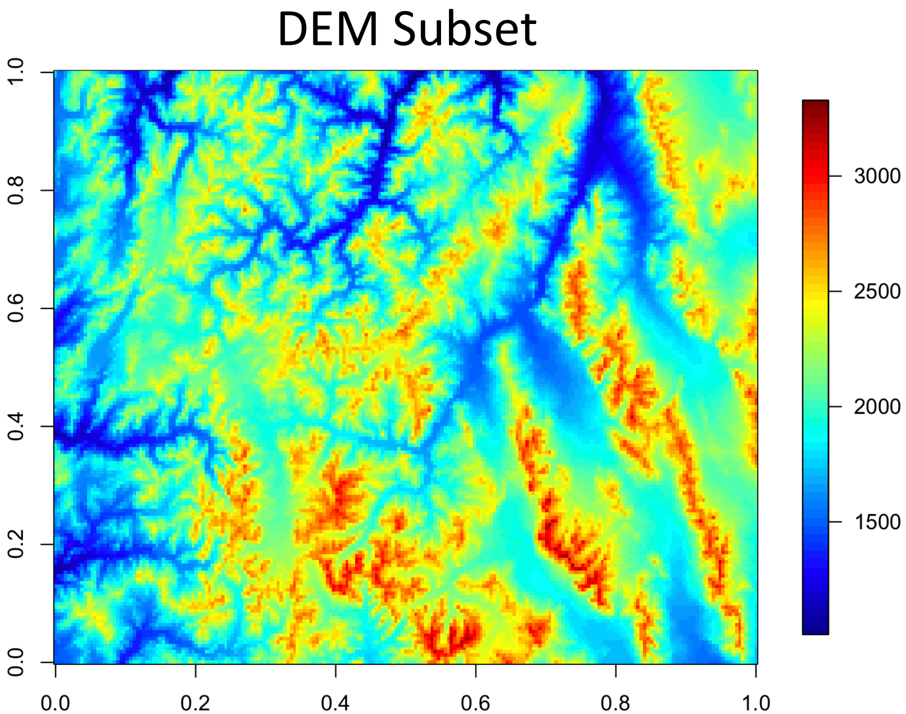
Networks are better represented with two step stream processing approach

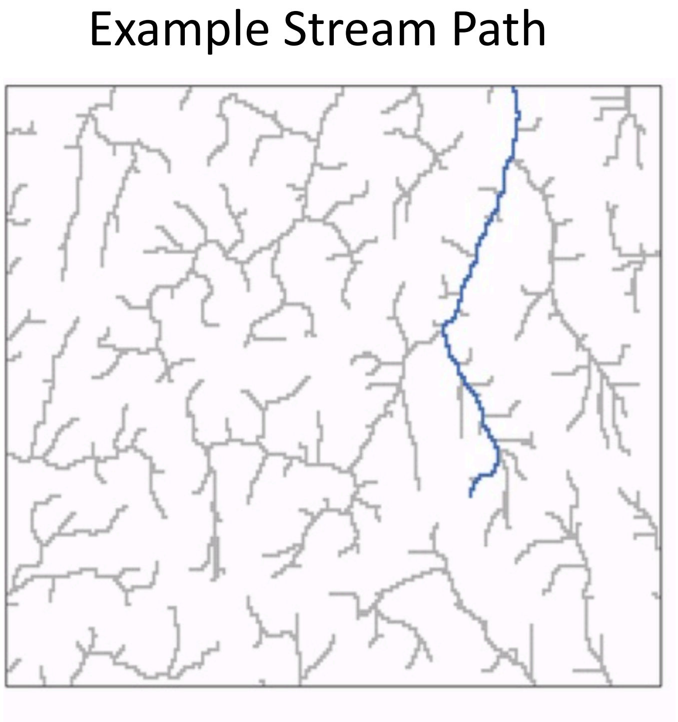
Area improvements using 5th order streams



Step 2: Apply smoothing along river network and calculate slopes for ParFlow

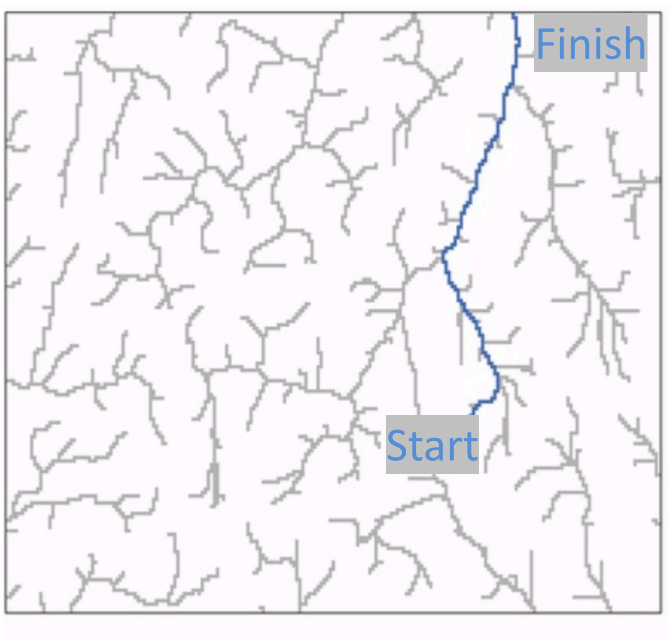
Without some smoothing stream slopes may be very noisy

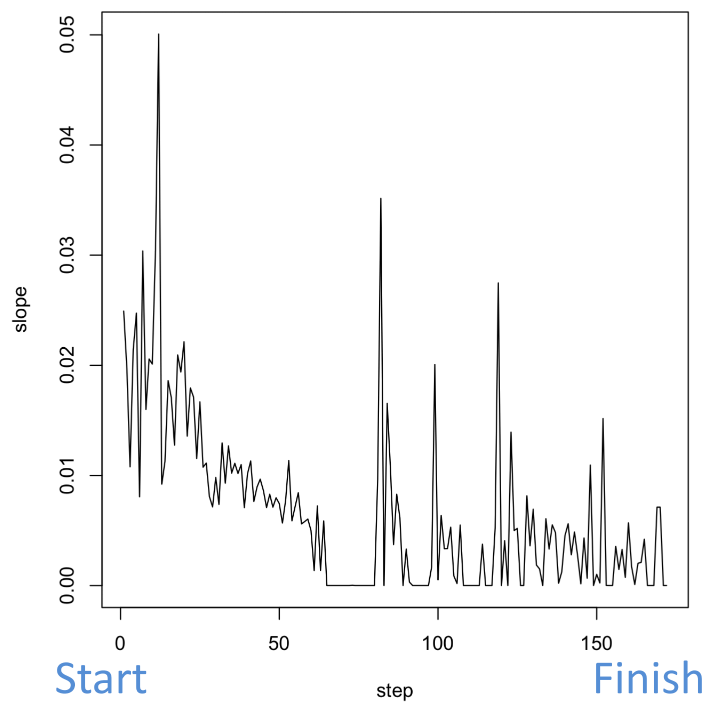




Without some smoothing stream slopes may be very noisy

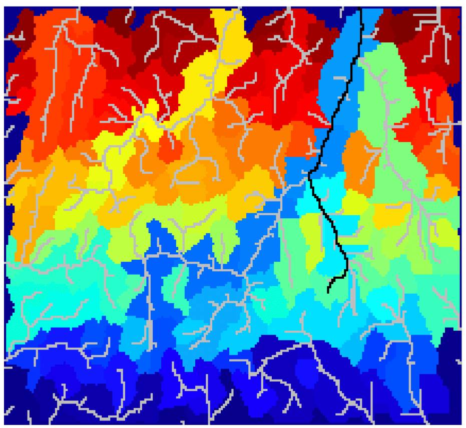
Example Stream Path



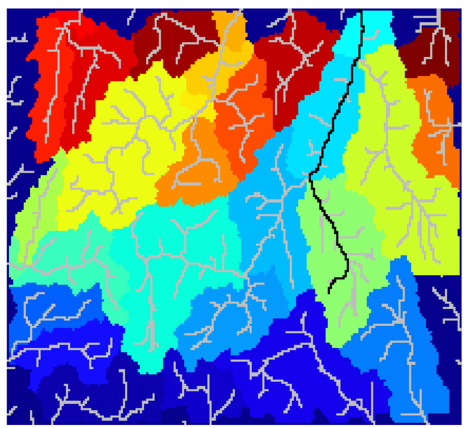


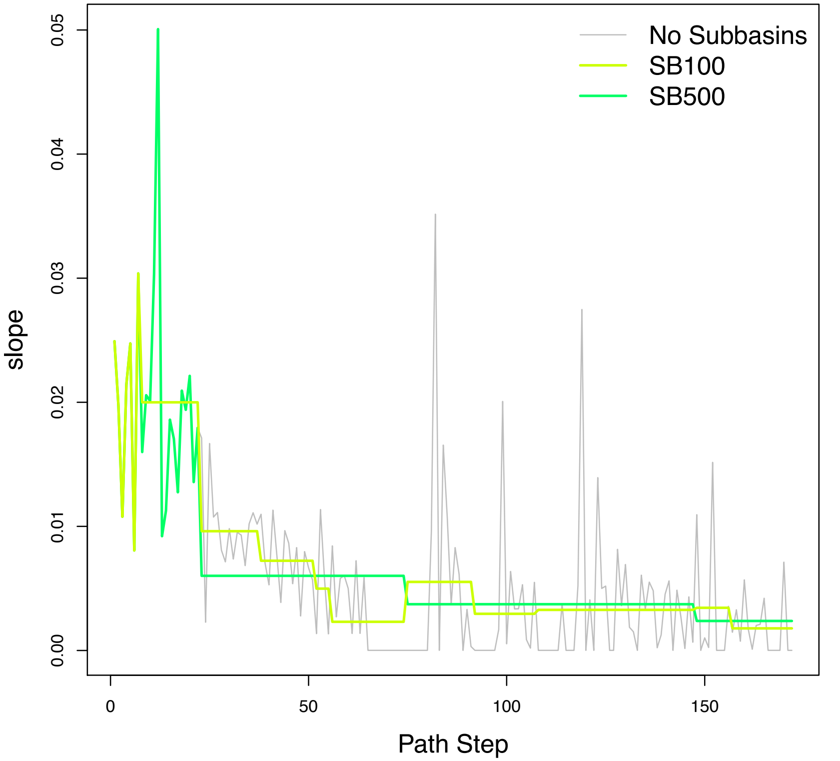
Smoothing applied along stream reaches using area thresholds

Sub basin Area Threshold = 100



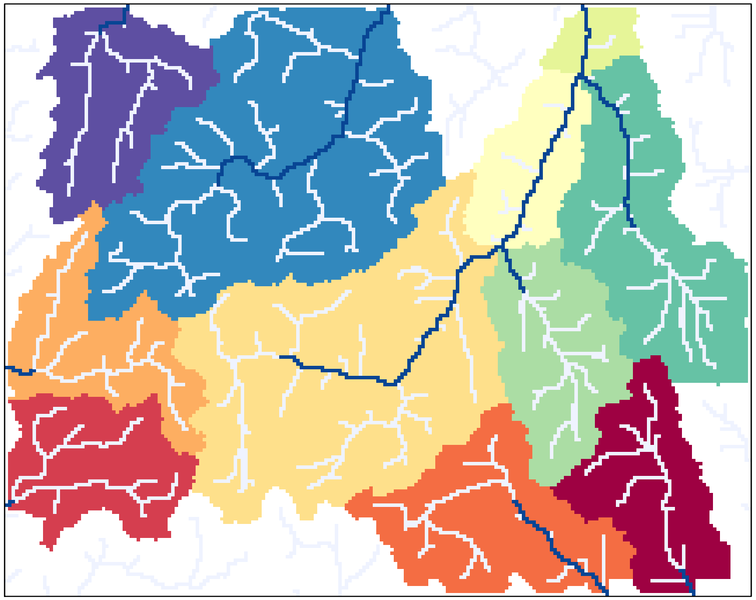
Sub basin Area Threshold = 500

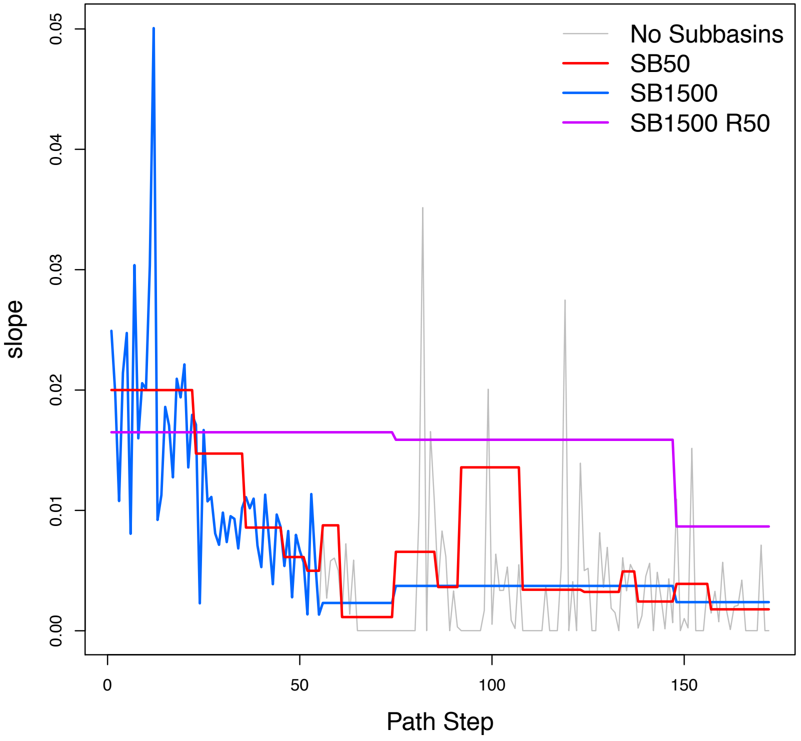


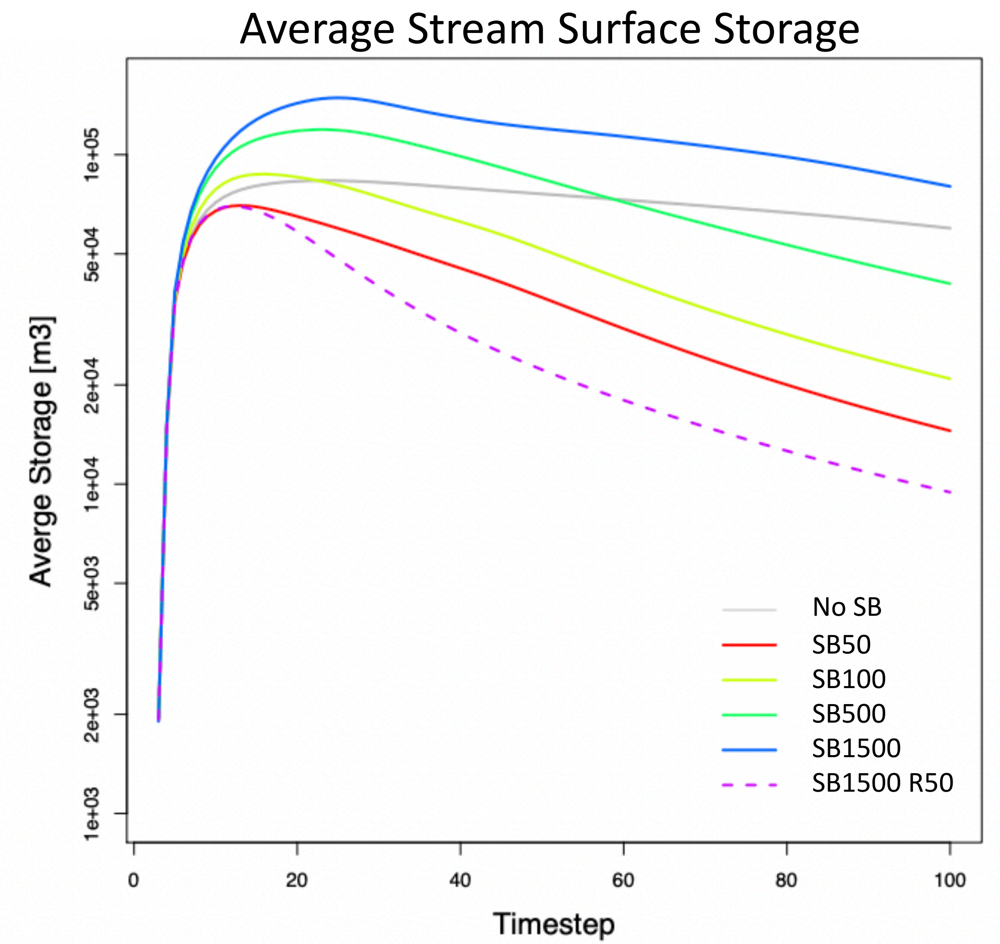


Smoothing applied along stream reaches using area thresholds

Sub basin Area Threshold = 1500

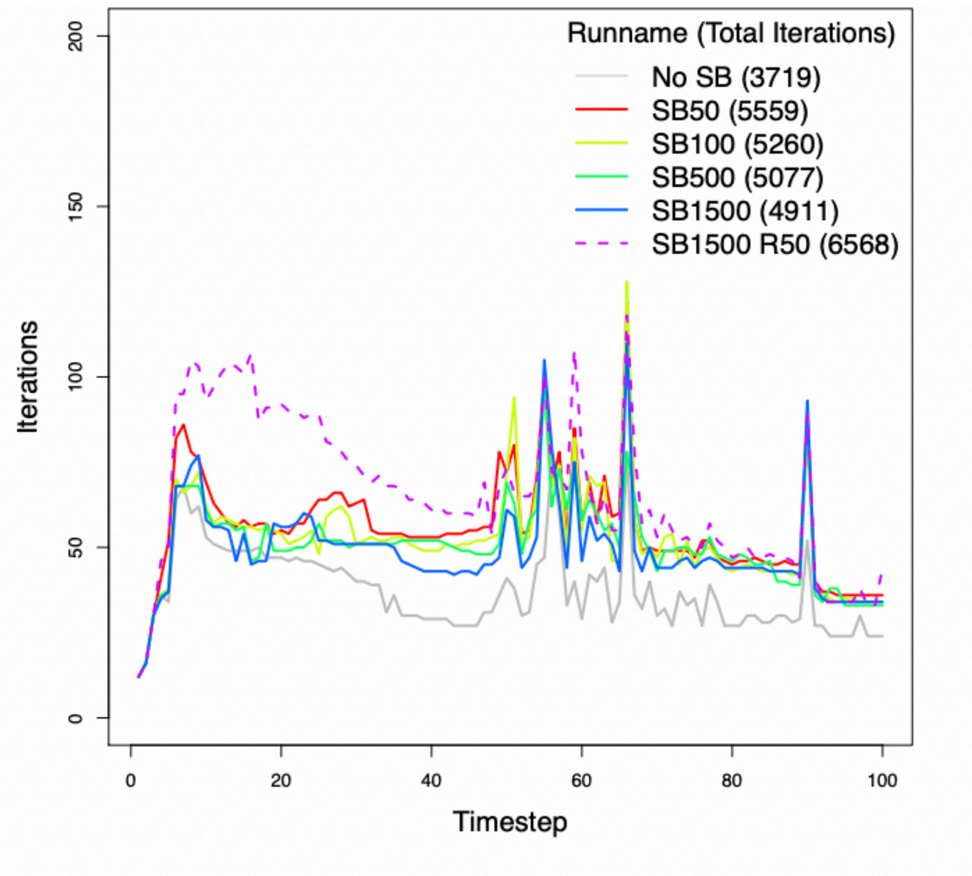






- Smoothing increases the runoff speed
- Decreasing the subbasin area threshold increases the slope variability but applies smoothing to more of the domain

Number of solver iterations



- Smoother domains increase solver iterations due to sharper streamflow response

Topographic processing exercise