SPAN flow models implemented in ARIES

Gary W. Johnson, Kenneth J. Bagstad

Version 1.0

April 2012

Table of Contents

1. Overview
2. Carbon sequestration and storage
   2.1. Overview
   2.2. Procedure
   2.3. Outputs
3. Aesthetic viewsheds
   3.1. Overview
   3.2. Procedure
   3.3. Outputs
4. Surface water supply
   4.1. Overview
   4.2. Procedure
   4.3. Outputs
5. Sediment regulation
   5.1. Overview
   5.2. Procedure
   5.3. Outputs
6. Riverine flood regulation
   6.1. Overview
   6.2. Procedure
   6.3. Outputs
7. Subsistence fisheries
   7.1. Overview
   7.2. Procedure
   7.3. Outputs
8. Open space proximity
   8.1. Overview
   8.2. Procedure
   8.3. Outputs
9. Coastal flood regulation
   9.1. Overview
   9.2. Procedure
   9.3. Outputs

1. Overview

In this document, we describe eight implementations of the Service Path Attribution Network (SPAN) framework, one per ecosystem service currently modeled by the ARIES system[1]. Each one

1 Further technical documentation on the flow algorithms is provided by Johnson, G.W. et al. (in press) Service Path

is given up to three matrices with the same numbers of rows and columns, which represent the continuous, spatial distribution of three variables – source, sink, and use values, described below – over the same geographic extent. These represent features relevant to the type of ecosystem service-specific matter, energy, or information – called the service medium – whose movement through space is the purpose of the given SPAN implementation.

The first matrix shows the distribution of the service medium's **source**. That is, each cell in the matrix records the amount of the medium that should be generated by this geospatial location over the course of the simulation.

The second matrix contains the distribution of the landscape's absorption capacity with respect to the service medium in the same units as the source matrix. We call these values the ecosystem service **sink**. Sinks exist for many but not all services, so the sink matrix will not exist in some instances.

The third matrix depicts the distribution of human populations or assets that are likely to be affected, for better or worse, by interaction with the service medium and may be either represented as a simple binary presence/absence matrix or with varying demand or vulnerability values per cell in the same units as the source matrix. This matrix is called the ecosystem service **use**. In the event that the service medium is beneficial to people (e.g., drinking water, pollen for crops, scenic views, i.e., "provisioning services"), **use** locations can be seen as potential beneficiaries of an ecosystem good or service. If the service medium is detrimental to people (e.g., floodwater, wildfire, storm surges, i.e., "preventive services"), **use** locations will represent populations or assets that are potentially vulnerable to an ecosystem-generated threat.

At the start of each SPAN simulation, one service carrier object is created in each source location with an initial weight equal to the value in the source matrix at that location (a density) times the area of that cell, so that each service carrier moves a volume around the matrix.

In order to determine each carrier's path through the matrix, each of the eight models below may require additional matrices. For example, the surface water supply, sediment regulation, and riverine flood regulation models all require matrices containing the spatial distribution of elevation and hydrologic networks, as well as floodplain extents data for sediment and riverine flood regulation models. The variables needed for service carrier routing varies from model to model and will be described in each of the sections below as needed.

Over the course of each simulation, these carriers will move from cell to cell through the matrix space, until either they become stuck in one cell permanently, they move beyond the bounds of the matrix's edges, or their weight falls below the **transition threshold**. This value, given in the same units as the **source** matrix, is the minimum amount of the service medium that the model user is interested in keeping track of. Thus, any service carriers with weights below this value are moving an insignificantly small amount of the medium around and will be disregarded by the simulation from there on.

Whenever a carrier encounters a **sink** location, it will give up some of its weight to that **sink** based on the absorption capacity still remaining in the cell at that point in the simulation. After this new absorption, the remaining **sink** value will be reduced by the carrier weight absorbed.

Whenever a carrier encounters a **use** location, a new carrier is stored in that cell, containing the entire route taken from the original **source** cell as well as a list of all the **sinks** encountered along

---

Attribution Networks (SPANs): A network flow approach to ecosystem services assessment. Forthcoming in: International Journal of Agricultural and Environmental Information Systems. Detailed descriptions of data and models to quantify sources, sinks, and uses of ecosystem services modeled in the ARIES system is provided by Bagstad, K.J. et al. (2011). Artificial Intelligence for Ecosystem Services (ARIES): A guide to models and data. Version 1.0 Beta. The ARIES Consortium: Bilbao, Spain.

this path and the amount of the carrier's weight that was absorbed by each of them. Finally, this stored carrier is given a weight, representing the amount of the service medium that affects this location for good or ill.

If the use process is destructive (i.e., typically rival use, such as water taken from streams to drink, thereby reducing the amount of downstream water available), then the stored carrier's weight will be equal to the remaining **use** value in this location. As with **sink** locations, this **use** value will be reduced by the weight captured from the service carrier in this step. Similarly, the moving service carrier's weight is reduced by the stored amount prior to moving on to the next cell in the matrix.

If the use process is non-destructive (e.g., floodwater encounters one **use** location, damages its assets, and continues on to cause damage to other **use** locations without the service carrier losing water to these encounters), then the stored carrier's weight will be equal to the full remaining weight of the moving service carrier. This carrier will then move on to the next cell in the matrix without losing any of the weight with which it entered the **use** location.

The SPAN simulation may be considered complete once all the moving service carriers have been terminated according to the conditions set forth above. At this point, all of the stored carriers are collected from the **use** locations and passed to a set of sixteen post-simulation analysis functions. Each of these functions produces a new matrix, containing the spatial distribution of one of the following variables:

Component	Specifier	Relation to Service Medium
Source	Theoretical	Expected production at origin
Source	Actual	Amount produced which reaches a use location
Source	Possible	Same as Actual Source but excludes sink effects
Source	Inaccessible	Difference between Theoretical and Possible Source
Source	Blocked	Difference between Possible and Actual Source
Sink	Theoretical	Expected absorption capacity at origin
Sink	Actual	Amount absorbed along simulated flow paths
Sink	Inaccessible	Difference between Theoretical and Actual Sink
Use	Theoretical	Expected demand/vulnerability at origin
Use	Actual	Amount received along simulated flow paths
Use	Possible	Same as Actual Use but excludes sink effects
Use	Inaccessible	Difference between Theoretical and Possible Use
Use	Blocked	Difference between Possible and Actual Use

Component	Specifier	Relation to Service Medium
Flow	Actual	Amount delivered to a use location
Flow	Possible	Same as Actual Flow but excludes sink effects
Flow	Blocked	Difference between Possible and Actual Flow

These values are given as densities in their respective matrices so as to be commensurate with the original input **source**, **sink**, and **use** matrices.

In order to run in conditions of data scarcity, all of the SPAN implementations below were created to accept input matrices with either deterministic values or probability distributions per cell, depending on the quality and completeness of the input data set. For all ecosystem services but carbon sequestration and storage, the SPAN model uses variance propagation to transmit the input uncertainties through to the outputs. In the special case of carbon, in which all **source** locations are connected to all **use** locations, the SPAN model uses a Monte Carlo simulation approach.

2. Carbon sequestration and storage

2.1. Overview

The carbon SPAN model computes the mass of carbon sequestered and stored that is available to offset anthropogenic carbon emissions produced within the same region. While computing carbon sequestration and storage may be sufficient for many applications, the identification of flow paths allows users to compute regional carbon budgets by interpreting human carbon emitters as users of carbon sequestration, with carbon-emitting ecosystems as sinks in the flow model. Because all source locations (carbon sequestering ecosystems) are connected to all sink and use locations by fast atmospheric mixing, the standard SPAN approach of tracking explicit routes from source to use locations is not adopted here. Instead, the algorithm simply distributes the remaining source value from each location among all use locations based on their relative emissions values.

2.2. Procedure

1. The model first computes the cell size in hectares for sources, sinks, and users given cell-width and cell-height. Multiplying this value by the densities (tons of CO_2 per hectare) in each cell of the source, sink, and use matrices calculates the tons of CO_2 sequestered, released, or emitted per cell.
2. Next, the user selects a number of iterations for the Monte Carlo simulation and draws this many sample worlds (e.g., one sample matrix each for source, sink, and use) by drawing sample values from the probability distributions in each of the matrix cells.
3. For each such sample world, the model:
    1. Calculates the total source, sink, and use values by summing across all cells in each matrix.
    2. Determines the total amount each source cell loses to all the sinks by multiplying each source value by the total sink divided by the total source. If the total sink is greater than the total source (i.e., there are more landscape emissions than sequestration), then every source cell simply loses all of its value to the sinks. Subtracting this result from the source value per cell gives the source amount remaining, which can be contributed to

users (i.e., used to offset anthropogenic emissions). This can be called the **unsunk source** value.

3. Computes the total amount each sink reduces all the sources by multiplying each sink value by the total source divided by the total sink. If the total source is greater than the total sink, then the amount sunk by every sink cell is simply its sink value. This is the **actual sink** value.
4. Computes the total amount each user receives from all the sources by multiplying each use value by the total source divided by the total use. If the total source is greater than the total use, then the amount received by every use cell is simply its use value. This is the **possible use** value.
5. Repeats step 4 above but replaces the source values with the unsunk source values from step 2. This computes the **actual use** value.
6. Normalizes all source and use values between [0-1] by dividing by their totals.
7. Pairs every source cell with every sink cell, and for each such pair, multiplies the normalized source value by the actual sink value. This calculates the amount each source cell loses to each sink cell.
8. Pairs every source cell with every use cell, and for each such pair, multiplies the normalized source value by the possible use value. This calculates the possible amount each source cell contributes to each use cell.
9. Pairs every source cell with every use cell, and for each such pair, multiplies the normalized source value by the actual use value. This calculates the actual amount each source cell contributes to each use cell.
10. For each triplet combination of source, sink, and use cells, multiplies the normalized source value by the normalized use value and then by the actual sink value to compute the effect of each sink on each pair of source and use point.

4. Combines the results of each Monte Carlo iteration from step 3 above by computing a distribution per use cell of the possible and actual service medium amounts received as well as the amounts lost to sinks.
5. Finally, divides the possible-weight, actual-weight, and sink-effects values by the number of hectares per cell to turn them back into densities (tons of CO2 per hectare per year) for comparison with the original input layers.

2.3. Outputs

The carbon model produces theoretical, possible, actual, inaccessible, and blocked carbon sources (sequestration), sinks (stored carbon release), and uses (anthropogenic emissions). All values are in tons of CO2 per hectare per year.

Subtracting the total theoretical sink value (summed across all cells and multiplied by the number of hectares per cell) from the total theoretical source value (summed across all cells and multiplied by the number of hectares per cell) calculates the tons of carbon sequestration per year available for offsetting anthropogenic emissions. The net carbon balance of the study region can then be determined by subtracting the total theoretical use value (summed across all cells and multiplied by the number of hectares per cell) from this available offset amount.

3. Aesthetic viewsheds

3.1. Overview

The aesthetic viewsheds SPAN model uses lines of sight to connect and quantify view paths between source locations (visually valued objects) and use locations (areas of potential enjoyment, such as housing), check for obstructions and sink features (visual blight), and determine, using a digital elevation model (DEM), how much of the source can be seen from a given use location. The source, sink, and use inputs give relative rankings for sources, sinks, or users of visually valued viewsheds. Sink features that can degrade viewsheds are accounted for only if they are present in the foreground of a user's view of a source location. A distance decay function is applied to compute the visual utility originating from the source location that reaches each user.

3.2. Procedure

For each source point-use point pair, given that the points are not in the same location (no *in situ* use occurs), the model:

6. Computes the Euclidean distance in meters between the center of the source and use cells.
7. Calculates the source decay at this distance, and proceeds if positive (Fig. 1). Otherwise, there is no view service between this source and use point pair.
8. Looks up the elevation at every point between the source and use locations.
9. Takes the difference in meters between each elevation along this view line and the use point's elevation (i.e., computes the rises).
10. Calculates the Euclidean distance in meters between every point along this view line and the use location (i.e., computes the runs).
11. Calculates each of the view obstruction slopes along the view line by dividing each rise by its corresponding run.
12. Selects the maximum view obstruction slope from these values to be the determinant of how much of the source a user can and cannot see along this view line.
13. Computes the view elevation in meters projected onto the source by multiplying the maximum view obstruction slope by the Euclidean distance between the source and use points in meters and adding this value to the use elevation.
14. If the projected view elevation is negative (because the view obstruction slope was negative), then sets it to 0.
15. If the slope is positive, then the user is looking up. The model only counts the visible part of the source value. If the projected view elevation is greater than or equal to the source elevation, then the user cannot see the source location due to intervening obstructions. Therefore the source value along this line of sight is 0. If the projected view elevation is less than the source elevation, then the user can see some fraction of the source location. The model subtracts the projected view elevation from the source elevation and divides this difference by the source elevation to get a number between 0 and 1, representing the visible source fraction. This fraction is then multiplied by the total source value to get the visible source value.
16. If the slope is negative, then the user is looking straight ahead or down (e.g., looking from a hill down at a lake). If the projected view elevation is greater than the source elevation, then the user cannot see it, and the visible source value is 0. Otherwise, if the projected view

elevation is less than or equal to the source elevation, then the user can see the entire source element. Thus, the model sets the visible source value to be the total source value.

17. Finally, multiplies the visible source value by the source decay at that user-to-source distance (Fig. 1) to compute the possible weight transmitted from the source point to the use point.
18. If the possible weight is below the transition threshold, then there is no service along this view line. Otherwise, continues with the next step.
19. For each sink location along the sight line, calculates the visible sink value using the same algorithm as for the source (e.g., projects view elevation onto the sink, multiplies sink value by fraction of sink elevation visible above the projected view elevation), and finally multiplies the visible sink value by the sink decay (Fig. 2) at its distance from the use point.
20. Computes the actual weight as the possible weight minus the sum of the sink effects along the sight line. If this value is negative (because sinks outweigh the source value), sets the actual weight to 0.
21. Stores a carrier on the use point with the possible and actual weights computed as described above, the line of sight route between the source and use point, and a list of sinks and their visual blight impacts on the actual weight value.

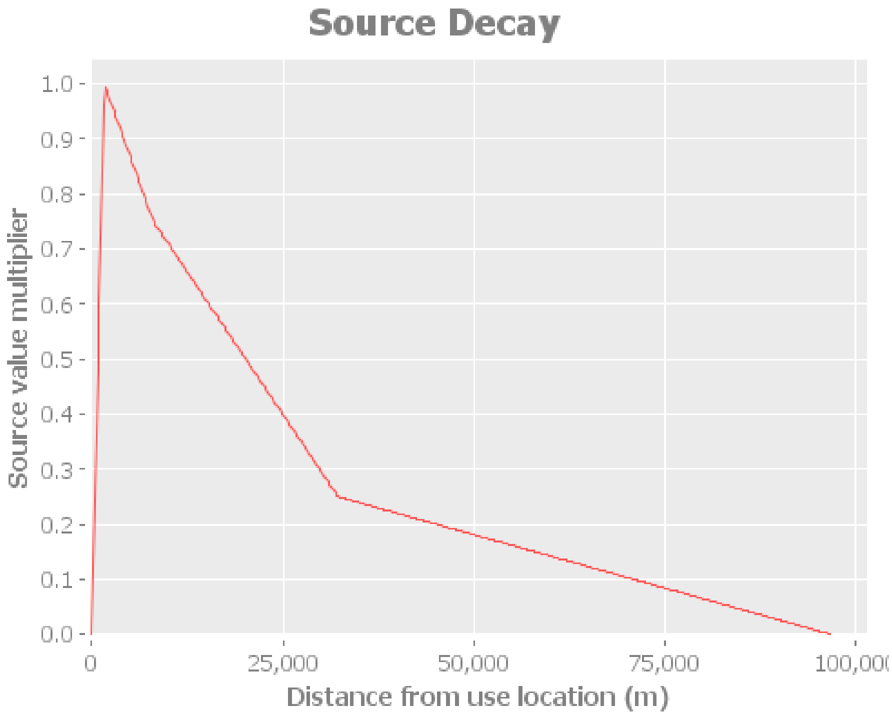

Figure 1: Source Decay: Grow from 0% to 100% over 1.6km. Next, reduce from 100% to 75% from 1.6 to 8km. Then, reduce from 75% to 25% from 8 to 32km. Finally, reduce from 25% to 0% from 32 to 96km.

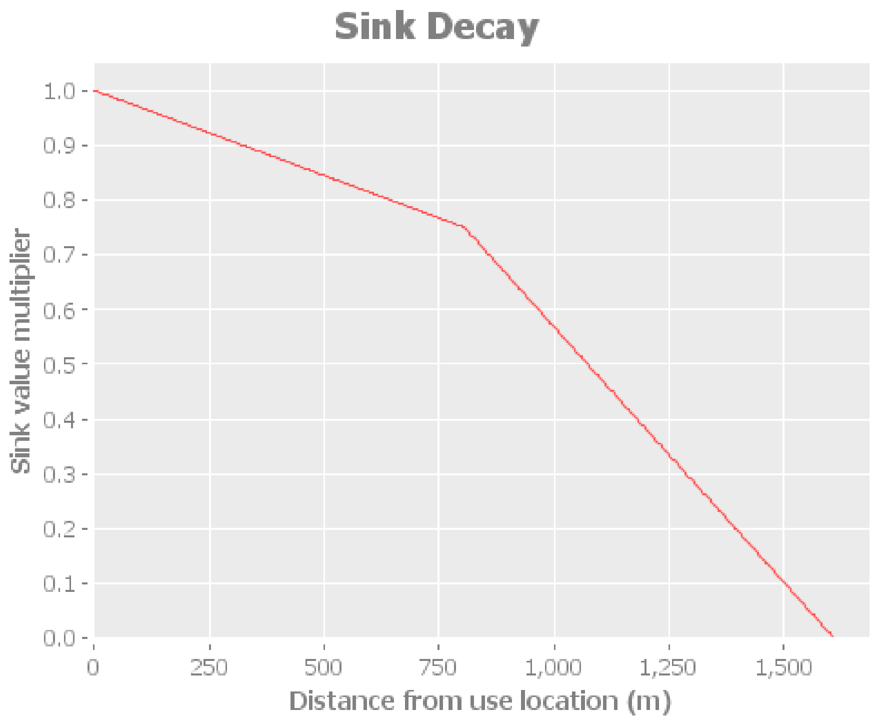

Figure 2: Sink Decay: Reduce from 100% to 75% over first 800m. Then reduce from 75% to 0% over the next 800m.

3.3. Outputs

The line-of-sight model produces theoretical, possible, actual, inaccessible, and blocked viewshed sources, sinks, uses, and flows. While the input source, sink, and use data were in relative values from 0-100, the flow model outputs can have much higher values. The reason for this is that the service medium moving between locations is scenic beauty transmitted by light, which is predominantly informational and is not subject to the laws of conservation of mass or energy. That is, if many users all have a view of the same mountain, they can all benefit from this view without competing with one another.

Therefore, since at best each source location could potentially provide a fully unobstructed, undecayed view to every user, the theoretical source matrix contains the input source values (from 0-100) times the number of use locations.

Since each sink could at worst impact every line-of-sight path once, the theoretical sink matrix contains the input sink values (from 0-100) times the number of line-of-sight paths possible (i.e., the number of source point-use point pairs).

Finally, since each user could at best have a fully unobstructed, undecayed view of every source point, the theoretical use map contains the sum of all the input source values (each from 0-100) assigned uniformly to every use location.

The most interesting outputs from this model are most likely to be the possible and actual source, use, and flow matrices as well as the actual and inaccessible sinks.

One known limitation of this model is that it assumes that the Earth is flat for the purposes of distance decay calculations. An improvement to the algorithm that incorporates the effects of the Earth's curvature on views at a distance is currently under development. We are also investigating the use of different distance decay functions for views of mountains (which may appear more impressive from a moderate than a close distance and likely have a slower distance decay function) than for other types of views (e.g., water bodies or visually significant vegetation), which may have a steeper distance decay function.

4. Surface water supply

4.1. Overview

The surface water supply SPAN model traces the path taken by runoff downhill and downstream according to a digital elevation model (DEM) and stream network data. The source matrix represents the total amount of runoff in mm per cell expected to be generated in each location over the time period of the simulation (one year in the initial ARIES release). The sink matrix contains the expected water absorption capacity of each cell (e.g., via evapotranspiration and soil infiltration) in mm per cell. Finally, the use matrix shows the spatial distribution of human demand for surface water also in mm per cell. As runoff carrier agents move from cell to cell during the simulation, their weight (the remaining runoff value) will be reduced by encounters with both sinks and users because the use process (i.e., extracting water from rivers and streams) reduces the amount available to downstream users in this case.

4.2. Procedure

22. The model first calculates the cell size in mm^2 using the cell width and cell height and multiplies this by the input source, sink, and use values as well as the transition threshold to convert them from densities (mm) to volumes (mm^3).
23. Finds the nearest river or stream cell to each user using a Euclidean distance metric. In cases of low resolution input data, some users may appear to be co-located with river or stream cells, in which case this will be considered the nearest one to them. Multiple users may share the same nearest river or stream cell. This information is meant to identify the locations from which users withdraw water from rivers or streams.
24. Places a virtual bucket at each sink point in the landscape, whose capacity is the sink volume (in mm^3) at that location. This is called the **sink capacity**.
25. Places two virtual buckets at each withdrawal point detected in step 2, whose capacities are equal to the sum of all the use volumes in their nearest use locations (as determined by step 2). One bucket is called the **possible use capacity**, and the other is called the **actual use capacity**. The first keeps track of the amount of surface water available to users at that location if there were no sinks on the landscape, and the second tracks the surface water available in the presence of sinks.
26. Creates one runoff service carrier per source cell, and sets its initial possible and actual weights to the source volume at that location.
27. Moves all service carriers one step either downhill (if overland) or downstream (if in a river or stream cell). This is a two-step process:
    1. If overland and in a sink location whose sink capacity bucket is not already full of water from previous carriers, reduces the carrier's actual weight by the remaining sink capacity and reduces the remaining sink capacity by the amount of water absorbed. Records the amount absorbed by this sink in the carrier's sink effects list.

       If in a river or stream location that contains a possible or actual use capacity bucket that is not already full, reduces the carrier's possible weight by the remaining possible use capacity, reduces the carrier's actual weight by the remaining actual use capacity, and reduces the remaining possible and actual use capacities by the respective amounts of water that they each absorb. Stores a new runoff carrier on this location with possible and actual weight values equal to the amounts of water absorbed by this location divided by the mm^2 per cell value (to translate them back into densities for comparison with the input matrices).

2. If the runoff carrier's possible weight is greater than the transition threshold volume, it moves to the neighboring cell with the lowest elevation, excluding the previous cell from which the carrier just came (to avoid short infinite cycles due to locally flat topography). If more than one neighboring cell meets this criteria, selects the one for which the angular distance is smallest between the current bearing and that required to reach this neighbor. Because this calculation cannot be performed on the matrix edges, any carrier which reaches an edge cell will be considered to have left the bounds of the study area.

28. Repeats step 6 until all service carriers have either gotten trapped in a single cell, flowed off the edge of the matrix bounds, or lost all of their possible weight to users.
29. Distributes the water captured in all the possible and actual use capacity buckets (those placed at stream withdrawal locations) to their nearest users (as determined by step 2) according to their relative use values. That is, use locations with greater use values (i.e., surface water demands) receive correspondingly more of the water captured at the stream withdrawal locations. Future versions of the model will be capable of accounting for legal water rights (i.e., ensuring that use requirements of senior water rights holders are accounted for first).

4.3. Outputs

The surface water model produces theoretical, possible, actual, inaccessible, and blocked surface water sources, sinks, uses, and flows. All values are in mm of water per cell per time period of the simulation (one year in the initial ARIES release).

5. Sediment regulation

5.1. Overview

The sediment regulation SPAN model traces the path taken by eroded soil downhill, downstream, and onto floodplains according to a DEM, stream network, and floodplain extents data. The source matrix represents the total amount of eroded soil in tons per hectare expected to be lost from each location over the time period of the simulation (one year in the initial ARIES release). The sink matrix contains the expected soil capture capacity of each cell (e.g., via floodplain deposition) in tons per hectare. Finally, the use matrix shows the spatial distribution of human populations or assets that would either benefit from or be harmed by sediment deposition (as presence/absence values). As sediment carrier agents move from cell to cell during the simulation, their weight (the remaining soil amount they carry) will be reduced by encounters with sinks (but not users). Users that are co-located with sinks in floodplains will be affected by however much sediment is captured by their sink. All other users will receive no service according to this model.

5.2. Procedure

30. The model first calculates the cell size in hectares using the cell width and cell height and multiplies this by the input source, sink, and use values to convert them from densities

(tons/ha) to volumes (tons).

31. Finds the nearest river or stream cell to each sink in a floodplain using a Euclidean distance metric. In cases of low resolution input data, some sinks may appear to be co-located with river or stream cells, in which case this will be considered the nearest one to them. Multiple sinks may share the same nearest river or stream cell. If there is a levee between a floodplain sink and its nearest river or stream cell, this sink will be considered nullified by the levee's water-channeling effect. This information is meant to identify the river or stream locations from which sediment emerges onto floodplains during flooding events, when it can be deposited.
32. Calculates the relative position of each non-levee-blocked floodplain sink along a line between its nearest river or stream point and the floodplain boundary. This position value is a real number between 0 and 1, where 0 indicates the distant floodplain boundary and 1 indicates the river or stream point. These values are called the **floodplain activation factors** and indicate each sink's relative likelihood of sediment deposition within the floodplain.
33. Places a virtual bucket at each sink point not in a floodplain, whose capacity is the sink volume (in tons of sediment) at that location. This is called the **overland sink capacity**.
34. Places a virtual bucket at each sediment deposition point detected in step 2, whose capacities are equal to the sum of all the non-levee-blocked floodplain sink volumes in their nearest sink locations (as determined by step 2) times their respective floodplain activation factors (step 3). This is called the **floodplain sink capacity**.
35. Creates one sediment service carrier per source cell and sets its initial actual weight to the source volume at that location. Its possible weight may be disregarded as there is no sediment deposition service provided by a landscape without sinks.
36. Moves all service carriers one step either downhill (if overland) or downstream (if in a river or stream cell). This is a two-step process:
    1. If overland and in a non-floodplain sink location whose overland sink capacity bucket is not already full of sediment from previous carriers, reduces the carrier's actual weight by the remaining overland sink capacity and reduces the remaining overland sink capacity by the amount of sediment absorbed. Records the amount absorbed by this sink in the carrier's sink effects list.

       If in a river or stream location that contains a floodplain sink capacity bucket that is not already full, reduces the carrier's actual weight by the remaining floodplain sink capacity and reduces the remaining floodplain sink capacity by the amount of sediment absorbed. Records the amount absorbed by this sink in the carrier's sink effects list. Stores a new sediment carrier on this location with possible and actual weight values equal to the amount of sediment absorbed by this location divided by the hectares per cell value (to translate them back into densities for comparison with the input matrices).
    2. Moves to the neighboring cell with the lowest elevation, excluding the previous cell from which the carrier just came (to avoid short infinite cycles due to locally flat topography). If more than one neighboring cell meets this criteria, selects the one for which the angular distance is smallest between the current bearing and that required to reach this neighbor. Because this calculation cannot be performed on the matrix edges, any carrier which reaches an edge cell will be considered to have left the bounds of the study area.
37. Repeats step 7 until all service carriers have either gotten trapped in a single cell, flowed off the edge of the matrix bounds, or lost all of their actual weight to sinks.

38. Distributes the sediment captured in all floodplain sink capacity buckets (those placed at stream deposition locations) to their nearest floodplain sinks (as determined by step 2) according to their relative sink values (after multiplying by their floodplain activation factors). That is, floodplain sink locations with greater sink values (i.e., soil capture capacities) weighted by their floodplain activation factors receive correspondingly more of the sediment captured at the stream deposition locations.
39. If any of these floodplain sinks are co-located with a user, then service is considered to have been transmitted to that user. Otherwise, no service flow occurs at that location.

5.3. Outputs

The sediment regulation model produces theoretical, actual, and inaccessible sediment sources, sinks, uses, and flows. All values are in tons of sediment per hectare per time period of the simulation (one year in the initial ARIES release).

6. Riverine flood regulation

6.1. Overview

The riverine flood regulation SPAN model traces the path taken by runoff downhill, downstream, and onto floodplains according to a DEM, stream network, and floodplain extents data. The source values represent the total expected runoff volume per location over the time period of the simulation. Sink values quantify the expected water absorption capacity of each location. Users are mapped as human settlements or other assets that could be harmed by floodwater. As floodwater carrier agents move from cell to cell, their weight (the remaining runoff value) is reduced by encounters with sinks, but not by users. Users in floodplains that are in the path of floodwater will be affected proportionally to the floodwater volume that reaches them. No other users will be harmed in this model.

6.2. Procedure

40. The model first calculates the cell size in mm^2 using the cell width and cell height and multiplies this by the input source, sink, and use values to convert them from densities (mm) to volumes (mm^3).
41. Finds the nearest river or stream cell to each sink and user in a floodplain using a Euclidean distance metric. In cases of low resolution input data, some sinks or users may appear to be co-located with river or stream cells, in which case this will be considered the nearest one to them. Multiple sinks and users may share the same nearest river or stream cell. If there is a levee between a floodplain sink or user and its nearest river or stream cell, the sink will be considered nullified and the user will be considered fully protected by the levee's water-channeling effect. This information is meant to identify the river or stream locations from which floodwater emerges onto floodplains during flooding events, when it can harm users.
42. Calculates the relative position of each non-levee-blocked floodplain sink and user along a line between its nearest river or stream point and the floodplain boundary. This position value is a real number between 0 and 1, where 0 indicates the distant floodplain boundary and 1 indicates the river or stream point. These values are called the **floodplain activation factors** and indicate each sink's relative likelihood of capturing floodwater and each user's relative likelihood of being inundated.
43. Places a virtual bucket at each sink point not in a floodplain, whose capacity is the sink

volume (in mm^3 of water) at that location. This is called the **overland sink capacity**.

44. Places a virtual bucket at each floodwater emergence point detected in step 2, whose capacities are equal to the sum of all the non-levee-blocked floodplain sink volumes in their nearest sink locations (as determined by step 2) times their respective floodplain activation factors (step 3). This is called the **floodplain sink capacity**.
45. Creates one floodwater service carrier per source cell and sets its initial possible and actual weights to the source volume at that location.
46. Moves all service carriers one step either downhill (if overland) or downstream (if in a river or stream cell). This is a two-step process:
    1. If overland and in a non-floodplain sink location whose overland sink capacity bucket is not already full of runoff from previous carriers, reduces the carrier's actual weight by the remaining overland sink capacity and reduces the remaining overland sink capacity by the amount of runoff absorbed. Records the amount absorbed by this sink in the carrier's sink effects list.

       If in a river or stream location that contains a floodplain sink capacity bucket that is not already full, reduces the carrier's actual weight by the remaining floodplain sink capacity and reduces the remaining floodplain sink capacity by the amount of floodwater absorbed. Records the amount absorbed by this sink in the carrier's sink effects list.

       If in a river or stream location that is the nearest such point to one or more users (as determined by step 2), stores a new floodwater carrier on this location with possible and actual weight values equal to the amount of floodwater reaching this location (before local floodplain sink effects are applied) divided by the mm^2 per cell value (to translate them back into densities for comparison with the input matrices).
    2. Moves to the neighboring cell with the lowest elevation, excluding the previous cell from which the carrier just came (to avoid short infinite cycles due to locally flat topography). If more than one neighboring cell meets this criteria, selects the one for which the angular distance is smallest between the current bearing and that required to reach this neighbor. Because this calculation cannot be performed on the matrix edges, any carrier which reaches an edge cell will be considered to have left the bounds of the study area.
47. Repeats step 7 until all service carriers have either gotten trapped in a single cell or flowed off the edge of the matrix bounds.
48. Distributes the floodwater from the carriers stored in step 7.1 (those placed at river or stream locations nearest to users) to their nearest floodplain users (as determined by step 2) according to their floodplain activation factors (from step 3). That is, floodplain use locations with greater floodplain activation factors (i.e., nearer to the river or stream within the floodplain) are inundated by correspondingly more of the floodwater passing through these locations.

6.3. Outputs

The flood mitigation model produces theoretical, possible, actual, inaccessible, and blocked floodwater sources, sinks, uses, and flows. All values are in mm of water per cell per time period of the simulation (one year in the initial ARIES release). Since data to support flood modeling in the initial ARIES release are at an annual time scale, results are best interpreted as maps of general flood vulnerability and potential flood regulation by ecosystems. Data or models that operate at finer temporal resolutions will be needed to model flooding and flood regulation on an event-by-event basis.

7. Subsistence fisheries

7.1. Overview

The subsistence fisheries SPAN model simulates the near-shore, rival fishing behavior of non-commercial fishermen located near major water bodies. Source locations record the fish biomass available over the time period of the simulation. Use locations identify fish-dependent settlements and assign them individual demand values in the same units as the source values. Roads and trails connect fishermen to their nearest viable fishing grounds. No sink effects are included in this model.

7.2. Procedure

49. The model first calculates the cell size in km^2 using the cell width and cell height and multiplies this by the input source and use values to convert them from densities (kg/km^2) to volumes (kg).
50. Sets up source caches at all source points, recording the fish supply at those locations in kg.
51. Finds the nearest path cell to each user and follows this path towards the shore until the nearest source cell (i.e., water location containing fish) is encountered. This marks the entrance to this user's fishing grounds.
52. Sends each fisherman into the water at their fishing ground entrance and starts them fishing simultaneously, depleting the local source caches in the process. Since the fish being caught are a rival resource, their acquisition by one fisherman prevents further fishermen from using them.
53. If a location does not contain enough fish to meet a fisherman's demand, it proceeds to fish in each neighboring water cell until either its demand has been met or all cells within the maximum fishing range (5km in the initial ARIES release) of the initial fishing point have been exhausted. This is meant to bound the fishing area to a reasonable distance beyond which subsistence fishermen are unlikely to travel. This range may be varied for different subsistence fishing conditions.
54. Runs the simulation until all fishermen have either acquired their use value in fish (i.e., met their demand) or exhausted the source caches within their fishing grounds.
55. Records flow paths along footpaths from the use points to their fishing grounds entrances, which contain the total amount of fish caught by each fisherman in kg/km^2 (translated to a density for comparison with the input source and use values). Records flow values in each water cell equal to the amount of fish caught in that location by any fisherman.

7.3. Outputs

The subsistence fisheries model produces theoretical, actual, and inaccessible sources, uses, and flows. All values are in kg/km^2 of fish per time period of the simulation (one year in the initial ARIES release).

8. Open space proximity

8.1. Overview

The open space proximity SPAN model assesses the accessibility of open spaces (e.g., parks, fields, water bodies) within walking distance of residential housing in urban and suburban areas. Source locations (open spaces) are quantified using relative rankings of their potential appeal to nearby residents. Use locations identify buildings whose property values could benefit from access to nearby open spaces. Because this ecosystem service flows along walking paths between the source and use points, sink values rank the degree to which intermediate locations along these paths reduce the appeal of these routes to users. For example, if a property has a short path to a park that crosses a highway or passes through a dangerous neighborhood, the value of the nearby park to this property's value may be decreased.

8.2. Procedure

56. The model first constructs a service carrier at each source point with initial possible and actual weights equal to the source value at that location (from 0-100).
57. If the source location is also a sink, subtracts its sink value from the carrier's actual weight. The sink's ability to affect other carriers that pass through it is unaffected by this.
58. If the source location is also a use point (due to low-resolution inputs causing open space and a building to appear to be co-located), stores a service carrier on this location with the post-sink possible weight, actual weight, and sink effects.
59. If its remaining actual weight is above the transition threshold, splits the initial service carrier into one carrier per neighboring cell and moves these child carriers into those locations, each tracking a different route from the source location. The parent carrier is then removed from the simulation.
60. Reduces each of the child carriers' possible and actual weights by the distance decay along the path taken from the source point (Fig. 3).
61. Each of these carriers again checks for the presence of sinks and users in its new location, losing actual weight to sinks and/or storing carriers on use locations (as in steps 2 and 3 above).
62. Expands the frontier (i.e., bounding box of cells on the grid) around all of the remaining carriers (those with distance-decayed possible weights above the transition threshold) by creating one carrier per cell in the frontier. Each such carrier chooses as its parent the adjacent carrier from the previous frontier with the greatest actual weight. Multiple carriers can share the same parent.
63. Reduces each of the new frontier carriers' possible and actual weights by the distance decay along the path taken from the source point (determined by their chosen parent carrier).
64. Continues to expand the frontier around each source point until distance decay drives the possible weights of all service carriers below the transition threshold.

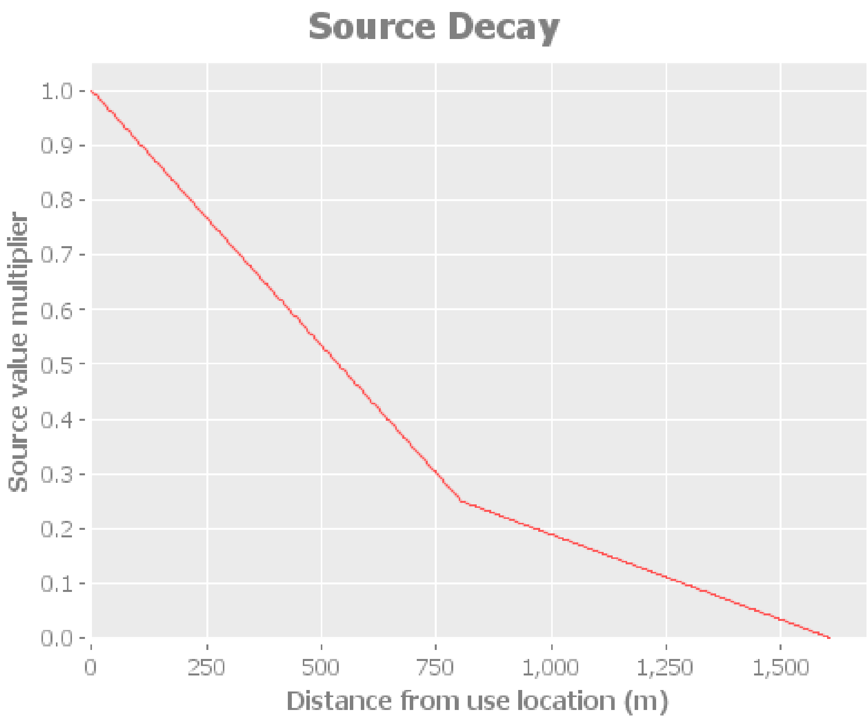

Figure 3: Source Decay: Reduce from 100% to 25% over first 800m. Then reduce from 25% to 0% over the next 800m.

8.3. Outputs

The open space proximity model produces theoretical, possible, actual, inaccessible, and blocked sources, sinks, uses, and flows. While the input source, sink, and use data were in relative values from 0-100, the flow model outputs can have much higher values. The reason for this is that the service medium moving between locations is predominantly an informational and aesthetic service provided to people along footpaths to open space locations. As a result, if many users all have a path to the same open space, they can all benefit from this location without competing with one another.

The aesthetic service provided by source locations, while non-rival, is potentially congestible (i.e., too many users in the same open space will eventually crowd it out to the point that its value is reduced). However, for the purposes of this model, this problem is simplified somewhat by ignoring the effects of congestibility. Future improvements to the algorithm will seek to take this issue into account.

Since at best each source location could potentially provide fully unobstructed, undecayed access to every user, the theoretical source matrix contains the input source values (from 0-100) times the number of use locations.

Since each sink could at worst impact every footpath once, the theoretical sink matrix contains the input sink values (from 0-100) times the number of footpaths possible (i.e., the number of source point-use point pairs).

Finally, since each user could at best have a fully unobstructed, undecayed path to every source point, the theoretical use map contains the sum of all the input source values (each from 0-100) assigned uniformly to every use location.

The most interesting outputs from this model are most likely to be the possible and actual source, use, and flow matrices as well as the actual and inaccessible sinks.

9. Coastal flood regulation

9.1. Overview

The coastal flood regulation SPAN model recreates the movement of a historical storm by tracing its storm track from a fixed distance offshore until it encounters vulnerable lives or assets. The service medium in this model is the energy of a storm surge, which is effectively measured in its height above sea level in meters. The source of the storm surge is chosen as a point along its historical storm track at a given distance from land and its source value is the storm surge height at that location. Sinks, which reduce the storm surge height as it moves, are both marine features (e.g., bathymetry, coral reefs, seagrass beds) and terrestrial characteristics (e.g. mangroves, dunes, coastal wetlands). Users are economic assets or lives at risk due to storm surge damage.

9.2. Procedure

For a given historical storm, the model:

65. Finds the point along its storm track that is a set *initial offshore distance* from land (100 km in the initial ARIES release).
66. Determines the direction that the storm is heading toward users by first calculating the mean bearing between the storm source point and all use locations. Then selects the neighboring storm track cell for which the angular distance is smallest between the mean bearing to users and that required to reach this neighbor. Because this calculation cannot be performed on the matrix edges, if a storm's source point begins on the edge, this simulation will be terminated immediately.
67. Calculates the cell size in m^2 using the cell width and cell height and multiplies this by the sink values and transition threshold to convert them from densities (m) to volumes (m^3).
68. Projects a line of carriers perpendicular to the storm direction that is *storm surge width* wide (100 km in the initial ARIES release). Each carrier's possible and actual weights are set to a volume of water (in m^3) equal to the source point's value (i.e., storm surge height in meters) times the area (in m^2) covered by projecting a line of cells behind it to a distance of *storm surge depth* (5 km in the initial ARIES release). These carriers represent the energy of the storm surge.
69. Traces the storm track backwards from the source point up to a distance of half the *maximum sample window size* (10 km in the initial ARIES release) and then traces it forwards for the distance reached backwards. This constructs a sample storm track window centered on the source point.
70. Calculates the mean bearing along the sample storm track window.
71. Moves the storm surge toward the coast by advancing the carriers in parallel with the storm surge centerpoint (the location which follows the storm track directly). To do this, the model:
    1. Shifts the sample storm track window forward one step, recalculating its mean bearing. If the leading edge of the window reaches the end of the storm track, shrinks the window from the trailing edge so that it remains centered on the storm centerpoint.
    2. Calculates the angular rotation between the current storm track bearing and the new mean bearing of the shifted sample storm track window. This determines how much and in what direction the storm is turning as is progresses along its historical storm track.

3. For each storm surge carrier, the model:
    1. Projects a line of cells from the carrier's current location to where it will be after the storm surge shifts forward and rotates (from step 7.2).
    2. Moves the carrier through each of the intermediate cells identified in the previous step, reducing its possible and actual weights by any sink values it encounters along the way.

       In this model, both possible and actual weights are reduced by geomorphic features (e.g., bathymetric slope, dunes, island topography). Actual weights are further reduced by ecological features (e.g., coral reefs, mangroves, seagrass, terrestrial vegetation).

       If at any point the carrier's possible weight falls below the transition threshold or it reaches the matrix edge, it is removed from the simulation, and its remaining weight is lost.
    3. Stores a carrier on any use location encountered with possible and actual weights equal to their values before local sink effects are applied divided by the m^2 per cell value (to translate them back into densities for comparison with the input matrices).

72. Repeats step 7 until either all of the carriers' possible weights have fallen below the transition threshold or the end of the storm track is reached.

9.3. Outputs

The coastal storm protection model produces theoretical, possible, actual, inaccessible, and blocked sources, sinks, uses, and flows for one historical storm track. All values are in meters of storm surge height above sea level.

Future improvements to this model will include more realistic treatment of shifts in storm surge direction. Although the model has been designed to run using historical storm track data (for which surge height data are available for calibration in some cases), the model could also be run using a simulated future storm or series of storms to generate a distribution of results under different hypothetical future storm conditions.