Fishing effect model - draft

T. Scott Smeltz

May 19, 2015

Contents

List of Figures

List of Tables

1 Introduction

We developed a time-varying, discrete time model to track fishing-related benthic habitat effects across the North Pacific. The Fishing Effects (FE) model accounts for spatially explicit historic fishing efforts, gear specific habitat susceptibility, and habitat specific recovery dynamics. The FE model is adapted from the Fujioka Long-term effect index (LEI) continuous time model (Fujioka, 2006; Sethi, Harris, & Rose, 2014). For parameter estimates, The FE model draws heavily on the literature review conducted for the Swept Area Seabed Impact (SASI) model (New England Fishery Management Council (NEFMC), 2011). Historic fishing efforts are measured using the Catch-in-Area (CIA) data set developed by NOAA (REF NEEDED). The CIA data contains the location of all commercial fishing activities since January 2003.

2 Model background

The FE model is conceptualized as an iterative model tracking habitat transitions between disturbed and undisturbed states. We let H represent the proportion of habitat disturbed by fishing activities, and h represent the proportion of habitat undisturbed by fishing activities. Terminology may vary slightly according to context, but in general, we will treat "undisturbed", "showing no effect of fishing" or other similar term as equivalent. In this model, habitat that has had no historic fishing is equivalent to disturbed habitat that has fully recovered. Likewise, we will treat terms such as "disturbed", "affected by fishing", or "impacted" as equivalent. We will often switch between the terms proportion and percentage when referencing the same parameter value. Generally, we will use proportions when discussing values as model inputs, and percentages when discussing these values in context.

The two habitat states, H and h are mutually exclusive and complete,

$$H + h = 1 \tag{1}$$

The FE model considers transition between H and h in monthly discrete time steps, t. Thus, H_t is undisturbed habitat and h_t is disturbed habitat at month t. In implementation of the model, $t = 1$ represents January 2003 when using the full historical fishing data. H transitions into h from one month to the next through fishing impacts and h transitions into H through recovery. We let I'_t represent the proportion of H that transitions to h by fishing impacts from month t to month $t + 1$, and ρ'_t is the proportion of h that recovers to H over the same time step. As a time-varying model, both I'_t and ρ'_t can vary from month to month. Thus, H_{t+1} is the is the sum of non-impacted H_t and recovered h_t. Conversely, h_{t+1} is the sum of impacted H_t and non-recovered h_t.

$$
\begin{aligned}
H_{t+1} &= H_t(1 - I'_t) + h_t\rho'_t \\
h_{t+1} &= H_t I'_t + h_t(1 - \rho'_t)
\end{aligned}
\tag{2}
$$

These state transitions are run independently within 5 km x 5 km grid cells across the North Pacific resulting in a spatially explicit tracking of H and h through time. In implementation of the model, we only track H since h can easily be back calculated through Eq. 1. Each grid cell is characterized by the proportion of five sediment types within it: mud, sand, granule/pebble, cobble, and boulder (see Section 3 for a discussion on the estimation of sediment proportions). Thus, a grid cell may be 50% sand and 50% mud, or 10% mud, 80% sand, and 10% cobble, or any other combination of sediment types that sums to 100%. Sediment types are assumed to be uniformly spread throughout each grid cell based on their proportion, thus this model does not consider spatial structure of sediment within a grid cell. H and h, then are tracked not only within grid cells, but also within sediment classes. Let the subscripts t, s represent grid time (month), and sediment class respectively. We will use a $\bullet$ to represent summations across a given dimension. Thus, the total undisturbed habitat in a given cell is the sum of undisturbed habitat for each sediment times the proportion of sediment with the grid cell, $P_{g,s}$, across all five sediment types (note the sediment proportion remains constant across all time periods).

$$
H_{t,g,\bullet} = \sum_{s=1}^{5} H_{t,g,s} P_{g,s} \tag{3}
$$

For example, if we have a grid cell composed of 10% mud, 80% sand, and 10% cobble, with H of 90%, 60%, and 100% for mud, sand and cobble respectively, the total undisturbed percent of the grid cell would be 67%. If we are interested in the proportion of disturbed habitat we could subtract $H_{t,g\bullet}$ from one (Eq. 1) or substitute h for H in Eq. 3. If we are interested in total undisturbed area within each grid cell at any given time, we simply need to multiply $H_{t,g,\bullet}$ times the the total area of the grid cell, A_g. The area for most grid cells will be 25 km^2 (5 km X 5 km), however, some grid cells will have smaller areas when they are located at the edge of the domain or along coastlines.

Both ρ' and I' Within each grid cell, I' is dependent on fishing effort and susceptibility of habitat by gear type and sediment. ρ' is dependent upon sediment type.

2.1 Fishing impacts

The proportion of undisturbed habitat that transitions to disturbed habitat as a result of fishing impact, I', is calculated as the exponentiation of the impact rate, I (for a discussion on this conversion, see Section 2.5),

$$I' = 1 - \mathrm{e}^{-}I \tag{4}$$

In the FE model implementation, the parameter I is indexed across grid cells, i, time periods, t, sediment classes, s, and gear types, g. We sum across n gear types to calculate an impact rate for each grid, time period, and sediment combination. For the remainder of the model discussion, we will omit the i and t indexing as all parameters are unique to grid cell and time period unless otherwise stated.

$$I_{s,\bullet} = \sum_{g=1}^{n} I_{s,g} \tag{5}$$

The impact rate for each gear-sediment combination, $I_{s,g}$, is calculated as the product of the gear specific fishing effort, f_g and the gear-sediment sensitivity $q_{s,g}$,

$$I_{s,g} = f_g q_{s,g} \tag{6}$$

f_g is a measure of the total bottom contact by each gear type as a proportion of the total grid cell area. It can range from zero, indicated no bottom contact by a gear type, to proportions greater than or equal one, indicating that the total bottom contact was greater than or equal the area of the grid cell. Proportions exceeding one may occur because f_g is summed across all individual tows of the same gear type within a cell regardless of possible overlap. When $f_g \geq 1$, it does not necessarily mean that the entire grid cell has been contacted by fishing gear, but only that the sum of bottom contact by individual tows is greater than or equal to the grid area. For example, we can consider the two following hypothetical (and unlikely) scenarios both resulting in $f_g = 1$. In one scenario, one tow may contact the the entire grid cell, resulting in 100% contact by one vessel. In a second scenario, 10 vessels may contact the same 10% area of the grid cell, in which case $f_g = 10 \times 0.1 = 1$. Although, $f_g = 1$ in both scenarios, the actual percent of ground contact differs (see Section 2.5 for more discussion on the implications of this difference). f_g is calculated as the nominal area swept by fishing gear, A_g multiplied contact adjustment, c_g. Nominal area swept is the door-to-door area of a tow not accounting for the degree to which the components of a tow actually touch the sea floor. The contact adjustment, then, is the proportion of the nominal area swept in contact with the sea floor. Because we assume a uniform distribution of sediment within a grid cell, f_g is not indexed over sediment, and is assumed to be spread proportionally among all sediments within a grid cell. Nominal areas are calculated for each tow, x, within a grid cell and are summed over n tows within gear types,

$$f_g = \frac{c_g \sum_1^n A_g}{A_i} \tag{7}$$

2.2 Estimate of sensitivity

Sensitivity, $q_{s,g}$, is the proportion of habitat affected by bottom contact with fishing gear. We index it over s and g because we assume differing sensitivities for gear-sediment combinations. Within each sediment class is a defined set of geological and biological habitat features. The sensitivity for a gear-sediment combination is the average of the sensitivity of all habitat features within a sediment class for a gear type. Habitat features definitions and their sensitivity were based on a literature review conducted for the SASI model (REFERENCE NEEDED). In a few cases, the SASI model split habitat feature sensitivity between high and low energy systems. IN these cases, we selected the low energy sensitivity. Habitat feature sensitivities were not estimated as absolute values, but were classified into four ranges: 0: 0–10%; 1: 10–25%; 2: 25–50%; 3: >50%.

To calculate an average sensitivity for each gear-sediment combination, we first randomly selected a sensitivity for each habitat feature within its range of sensitivities for a given gear-sediment combination. We then computed the mean of these randomly selected habitat feature sensitivities to get an average sensitivity for each gear-sediment combination. In the initial implementation of the FE model, we generated random sensitivities once then used these values throughout the model. In future version of the model, we may generate random sensitives for each time step and/or each grid cell.

Table 1: Sensitivity of habitat features to longlines

Feature Class	Feature	Mud	Sand	Gran-Peb	Cobble	Boulder
G	Bedforms		0			
G	Biogenic burrows	1	1			
G	Biogenic depressions	0	1			
G	Boulder, piled					0
G	Boulder, scattered, in sand					0
G	Cobble, pavement				0	
G	Cobble, piled				1	
G	Cobble, scattered in sand				0	
G	Granule-pebble, pavement			0		
G	Granule-pebble, scattered, in sand			0		
G	Sediments, suface/subsurface	0	0			
G	Shell deposits		0	0		
B	Amphipods, tube-dwelling	1	1			
B	Anemones, actinarian			1	1	1
B	Anemones, cerianthid burrowing	1	1	1		
B	Ascidians		1	1	1	1
B	Brachiopods			1	1	1
B	Bryozoans			1	1	1
B	Corals, sea pens	1	1			
B	Hydroids	1	1	1	1	1
B	Macroalgae			1	1	1
B	Mollusks, epifaunal bivalve, *Modiolus modiolus*	0	0	0	0	0
B	Mollusks, epifaunal bivalve, *Placopecten magellanicus*		0	0	0	
B	Polychaetes, *Filograna implexa*		1	1	1	1
B	Polychaetes, other tube-dwelling			1	1	1
B	Sponges		0	1	1	1

Adapted from the SASI model (NEFMC, 2011)
Sensitivity codes: 0: 0–10%; 1: 10–25%; 2: 25–50%; 3: >50%
Blank spaces are habitat features not associated with the given sediment class
G is Geological features and B is Biological features

Table 2: Sensitivity of habitat features to traps

Feature Class	Feature	Mud	Sand	Gran-Peb	Cobble	Boulder
G	Bedforms		0			
G	Biogenic burrows	1	1			
G	Biogenic depressions	1	1			
G	Boulder, piled					0
G	Boulder, scattered, in sand					0
G	Cobble, pavement				0	
G	Cobble, piled				1	
G	Cobble, scattered in sand				0	
G	Granule-pebble, pavement			0		
G	Granule-pebble, scattered, in sand			0		
G	Sediments, suface/subsurface	1	1			
G	Shell deposits		0	0		
B	Amphipods, tube-dwelling	1	1			
B	Anemones, actinarian			1	1	1
B	Anemones, cerianthid burrowing	1	1	1		
B	Ascidians		1	1	1	1
B	Brachiopods			1	1	1
B	Bryozoans			1	1	1
B	Corals, sea pens	1	1			
B	Hydroids		1	1	1	1
B	Macroalgae			1	1	1
B	Mollusks, epifaunal bivalve, *Modiolus modiolus*	0	0	1	1	1
B	Mollusks, epifaunal bivalve, *Placopecten magellanicus*		0	0	0	
B	Polychaetes, *Filograna implexa*		1	1	1	1
B	Polychaetes, other tube-dwelling			1	1	1
B	Sponges		0	1	1	1

Adapted from the SASI model (NEFMC, 2011)
Sensitivity codes: 0: 0–10%; 1: 10–25%; 2: 25–50%; 3: >50%
Blank spaces are habitat features not associated with the given sediment class
G is Geological features and B is Biological features

Table 3: Sensitivity of habitat features to trawls

Feature Class	Feature	Mud	Sand	Gran-Peb	Cobble	Boulder
G	Bedforms		2			
G	Biogenic burrows	2	2			
G	Biogenic depressions	2	2			
G	Boulder, piled					2
G	Boulder, scattered, in sand					0
G	Cobble, pavement				1	
G	Cobble, piled				3	
G	Cobble, scattered in sand				1	
G	Granule-pebble, pavement			1		
G	Granule-pebble, scattered, in sand			1		
G	Sediments, suface/subsurface	2	2			
G	Shell deposits		1	1		
B	Amphipods, tube-dwelling	1	1			
B	Anemones, actinarian			2	2	2
B	Anemones, cerianthid burrowing	2	2	2		
B	Ascidians		2	2	2	2
B	Brachiopods			2	2	2
B	Bryozoans			1	1	1
B	Corals, sea pens	2	2			
B	Hydroids	1	1	1	1	1
B	Macroalgae			1	1	1
B	Mollusks, epifaunal bivalve, *Modiolus modiolus*	1	1	2	2	2
B	Mollusks, epifaunal bivalve, *Placopecten magellanicus*		2	1	1	
B	Polychaetes, *Filograna implexa*		2	2	2	2
B	Polychaetes, other tube-dwelling			2	2	2
B	Sponges		2	2	2	2

Adapted from the SASI model (NEFMC, 2011)
Sensitivity codes: 0: 0–10%; 1: 10–25%; 2: 25–50%; 3: >50%
Blank spaces are habitat features not associated with the given sediment class
G is Geological features and B is Biological features

2.3 Recovery

Recovery, ρ'_s, is the proportion of disturbed habitat, h, that transitions to undisturbed habitat, H, from one time step to the next. We index it on sediment, s, because we assume differing recovery dynamics for different sediment classes. ρ' is calculated as the exponentiation of the negative recovery rate, ρ_s subtracted from one,

$$\rho'_s = 1 - \mathrm{e}^{-\rho_s} \tag{8}$$

ρ_s is defined as the inverse of recovery time,

$$\rho_s = \frac{1}{\tau_s} \tag{9}$$

where τ_s is the average number of years it takes for habitat in a sediment class to recover from a disturbed to an undisturbed state. In the implementation of the model, we divide ρ_s twelve to convert years to months (equivalent to multiplying τ_s by twelve) to align with the time step of the model. Similar to sensitivity, rho_s is calculated by averaging across all habitat features within a sediment class. However, we first average recovery times, τ, using he recovery times published for the SASI model (New England Fishery Management Council (NEFMC), 2011). We then convert average recovery times to recovery rate, rho_s using Eq. 9. Unlike the SASI model, which estimates a recovery time for each gear-sediment-habitat feature combination, the FE model does not account for recovery times differ when habitat is impacted by different gear types. Thus, when using the SASI values, we used their sediment-habitat features values for Trawls only, regardless of what gear caused the disturbance. In a few cases, the SASI recovery values differed for high and low energy systems. In these cases, we chose the low energy values. Also, like sensitivity, recovery times were classified into four ranges: 0: $<$ 1 year; 1: 1–2 years; 2: 2–5 years; 3: $>$5 years.

To calculate an average recovery time for each sediment class, we first randomly selected a recovery time for each habitat feature within its range of recoveries for a given sediment. We then computed the mean of these randomly selected habitat feature recoveries to get an average recovery time for each sediment class. We bounded class 3 to a maximum of ten years for recovery. In the initial implementation of the FE model, we generated random recoveries once then used these values throughout the model. In future version of the model, we may generate random recoveries for each time step and/or each grid cell. Additionally, it is worth noting, that in the current method of converting from yearly recovery rates to monthly recovery rates, we are assuming the recovery rate to be spread uniformly throughout the year. It is possible in future versions of the model to consider recovery rates that are seasonal or differ among months.

2.4 Expectation of recovery rate

The conversion in Eq. 8 is based on the exponential failure distribution (REF NEEDED).

2.5 Expectation of impact rate

We used Eq. 4 to convert impact rate, I to a proportion I' representing the proportion of undisturbed habitat that converts to disturbed habitat each time step. While I itself is measured as a proportion, it is calculated within each grid cell for each gear type by summing across the impacted area for each tow and dividing by the grid area. Because we sum across tows, regardless of whether or not they overlap, the value I can exceed of one. Using an untransformed I in the model would be problematic, as this could lead to estimations of disturbed area that exceed the total area of the grid cell. Eq. 4 solves this problem as the transformed I' is bounded between zero and one.

We can motive this particular transformation by imagining a grid cell to be composed of N discrete habitat units. We will consider an example with only one gear and sediment type in the grid cell. We will let n be the number of impacted habitat units impacted by fishing as summed across individual tows. Thus n is the product of I and N,

$$n = IN \tag{10}$$

n can exceed N if $I > 1$. Given only I as a measure of fishing activity, we don't know how much of the habitat was actually impacted. For example, if we imagine $N = 100$ discrete habitat units in a grid cell and $I = 1$, then $n = 100$. We don't know if all 100 units were impacted in the grid cell or if the same 10 units were impacted by 10 different tows ($I = 0.1$, for 10 tows). We can model this scenario by treating the impact of each unique tow a sampling *with replacement* from N discrete habitat features. For a habitat feature to be "sampled" means that it gets disturbed by fishing. We sample with replacement because each tow can disturb a habitat feature that has already been disturbed by another tow. We

Table 4: Recovery classifications for habitat features

Feature Class	Features	Mud	Sand	Gran-Peb	Cobble	Boulder
G	Bedforms		0			
G	Biogenic burrows	0	0			
G	Biogenic depressions	0	0			
G	Boulder, piled					3
G	Boulder, scattered, in sand					0
G	Cobble, pavement				0	
G	Cobble, piled				3	
G	Cobble, scattered in sand				0	
G	Granule-pebble, pavement			0		
G	Granule-pebble, scattered, in sand			2		
G	Sediments, suface/subsurface	0	0			
G	Shell deposits		2	2		
B	Amphipods, tube-dwelling	0	0			
B	Anemones, actinarian			2	2	2
B	Anemones, cerianthid burrowing	2	2	2		
B	Ascidians		1	1	1	1
B	Brachiopods			2	2	2
B	Bryozoans			1	1	1
B	Corals, sea pens	2	2			
B	Hydroids	1	1	1	1	1
B	Macroalgae			1	1	1
B	Mollusks, epifaunal bivalve, *Modiolus modiolus*	3	3	3	3	3
B	Mollusks, epifaunal bivalve, *Placopecten magellanicus*		2	2	2	
B	Polychaetes, *Filograna implexa*		2	2	2	2
B	Polychaetes, other tube-dwelling			1	1	1
B	Sponges		2	2	2	2

Adapted from the SASI model (NEFMC, 2011)
Recovery codes: 0: $<$ 1 year; 1: 1–2 years; 2: 2–5 years; 3: $>$5 years
Blank spaces are habitat features not associated with the given sediment class
G is Geological features and B is Biological features

can think of n as the number of times we take a sample with replacement of one from N. This assumes that there are n independent tows each with $I = 1/N$. Thus, each habitat feature has a $1/N$ probability of disturbance for each tow. Because a habitat feature can be repeatedly impacted, the probability of disturbance for each unit remains constant over all n tows. So, for any habitat feature, Y_i, the probability of being impacted k times follows a Binomial distribution, $\mathbf{Bin}(n, 1/N)$, with the probability mass function,

$$f(k; n, \frac{1}{N}) = \Pr(X_i = k) = \binom{n}{k} \frac{1}{N}^k (1 - \frac{1}{N})^{n-k} \tag{11}$$

Using Equation 11, we can calculate the probability of a habitat feature not impacted over n tows,

$$\Pr(X_i = 0) = (1 - \frac{1}{N})^n \tag{12}$$

Thus, the probability of a habitat feature being impacted is,

$$\Pr(X_i > 0) = 1 - \Pr(X_i = 0) = 1 - (1 - \frac{1}{N})^n \tag{13}$$

We can treat each X_i as a Bernoulli trail with the expectation of being impacted,

$$\mathbb{E}[X_i] = 1 - (1 - \frac{1}{N})^n \tag{14}$$

The expected proportion of impact I' across the entire grid cell will then be the sum of expected impacts for each habitat feature divided by N,

$$\frac{1}{N} \sum_{i=1}^{N} \mathbb{E}[X_i] = \frac{1}{N} N \mathbb{E}[X_i] = 1 - (1 - \frac{1}{N})^n \tag{15}$$

While Equation 15 models the grid cell and impact in discrete units, we can model these processes across a continuous surface by letting $N \to \infty$ and substituting IN for n using Equation 10,

$$I' = \lim_{N \to \infty} 1 - (1 - \frac{1}{N})^{IN} = 1 - \mathrm{e}^{-I} \tag{16}$$

We can interpret I' as the expected habitat disturbance, given an impact rate of I. Certainly, true measures of actual non-overlapping ground contact disturbance will vary about the expected value depending on how much overlap there is among tows. Likewise, we can anticipate higher variance as I increases, as greater impact will allow for greater variance in overlap patterns. We also note that the assumption of n independent tows each with $I = 1/N$, is almost certainly not met. Within a tow, impacts are not independent, and cannot be modeled as a sample with replacement since we know that individual tows do not overlap themselves (even where individual tows do intersect themselves, the area of the overlap is not counted twice). If a grid cell contained just one tow with an impact rate of $I = 0.25$, we know that the true proportion impacted is 25%. Using Eq. 4, however, we would estimate $I' = 1 - \exp(-0.25) = 0.22$, a difference of 0.03. This difference is small, and in general, $I' \approx I$ for low values of I (Fig. 1). For grid cell containing only a single tow, I will generally be small, as the width of a tow (max < 300 m) is small compared to the area of a typical grid cell (25 million sq. m). At greater values where we would expect multiple tows within a grid cell, I and I' do diverge considerably.

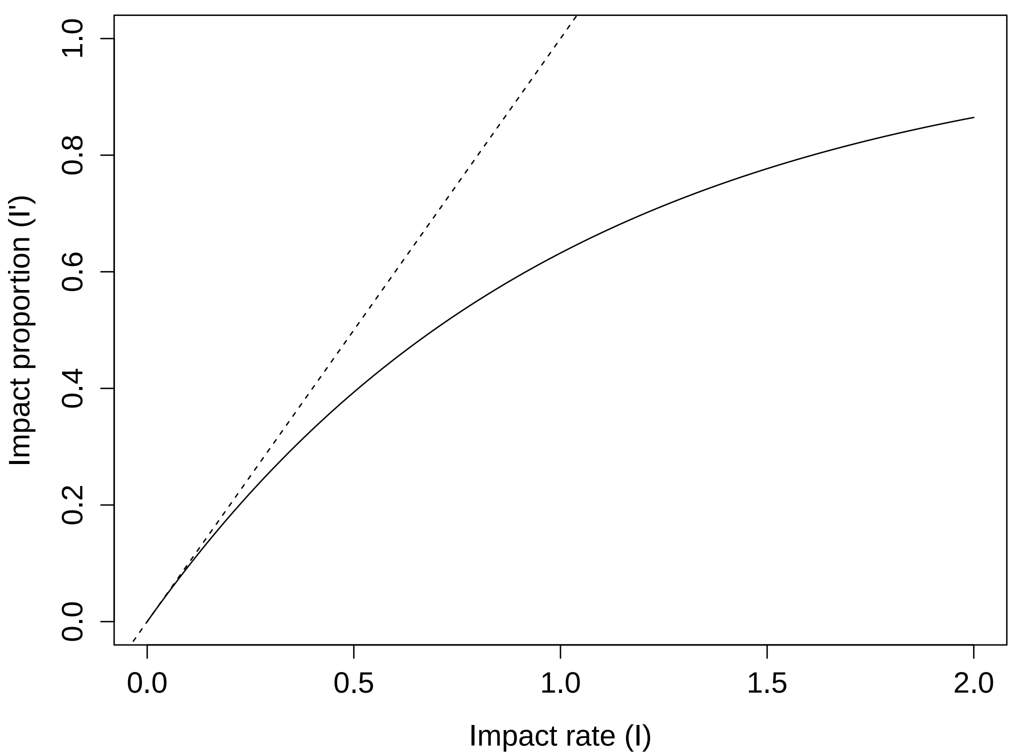

Figure 1: Impact rate, I, compared to impact proportion, I'
Solid curve shows $I' = 1 - e^{-I}$. Dotted line is one-to-one relationship

2.6 Calculation of fishing effort

Fishing effort, f_g is calculated for each cell, month, and gear type using the CIA data set (REF NEEDED). The CIA data set is provided as a polyline feature class representing individual tows. Each tow is attributed by three fields pertinent to the FE model: 1) CIA gear type 2) target species 3) tow date. For later calculation of gear-habitat feature sensitivity, the CIA gear types were simplified into gear types used in the SASI model (Table 5. Each tow was buffered based on its CIA gear type, target species and location combination. Locations were either Gulf of Alaska (GOA) or Bering Sea/Aluetian Islands (BSAI). A minimum and maximum nominal door width were determined (REF NEEDED) for each trawl combination (PTR and NPT). The buffer size was one half of the median of the minimum and maximum nominal door widths. For non-trawls (JIG, HAL, and POT), a buffer size was selected to account for an appropriate nominal area swept. The ArcMap (v 10.2.1) *Buffer* tool was used to create the buffers. Square buffer ends were used to ensure the area swept did not exceed the extent of the polyline as well as to increase the efficiency of subsequent spatial operations by reduced the number of vertices compared to a rounded buffer. The buffered tows were then intersected with the 5 km grid creating a nominal area swept for individual tows within each cell. Each of these nominal areas were multiplied by their contact adjustment to calculate total ground contact. Ground contacts for each FE model gear type were summed over each grid cell and month and divided by the grid cell area to calculate f_g.

Table 5: CIA gear compared to SASI gear

CIA gear	CIA gear code	SASI gear
Jig	JIG	Traps
Hook and line	HAL	Longline
Pots	POT	Traps
Pollock trawl	PTR	Trawl
Non-pelagic trawl	NPT	Trawl

Table 6: CIA species codes

Species Code	Target Species	Species Code	Target Species
A	Atka Mackerel	M	
B	Pollock - bottom	O	Other Species
C	Pacific Cod	P	Pollock - midwater
D	Deep Water Flatfish - GOA	R	Rock Sole - BSAI
E	Alaska Plaice	S	Sablefish - BSAI
F	Other Flatfish - BSAI	T	Greenland Turbot - BSAI
H	Shallow Water Flatfish - GOA	W	Arrowtooth Flounder
I	Halibut	X	Rex Sole - GOA
K	Rockfish	Y	Yellowfin Sole - BSAI
L	Flathead Sole		

Table 7: Fishery, gear, and buffer tables

CIA Gear type	Target species code	Location	BufferCode	Min width (m)	Max width (m)	Buffer Size (m)	Conatct adjust-ment
JIG	all	any	JIG			0.1	1
HAL	all	any	HAL	1	1	0.5	1
POT	all	any	POT			0.5	1
PTR	all	BSAI	PTR	100	180	70	1
PTR	all	GOA	PTR.G	33	146	45	1
NPT	K,A	any	NPT.R	40	50	22	1
NPT	C, P, B	GOA	NPT.GC	136	230	91.5	1
NPT	C, P, B	BSAI	NPT.C	100	100	50	1
NPT	D, E, F, H, L, R, T, W, X, Y, S, O	GOA	NPT.GF	183	265	91.5	1
NPT	D, E, F, H, L, R, T, W, X, Y, S, O	BSAI	NPT.BF	100	200	75	1

3 Sediment

We compiled sediment data from X surveys from the North Pacific. Data was points with sediment description. The surveys varied widely in methodology and sediment descriptions. The distribution of sediment points varied widely (Fig. 2). Most of the Continental shelf contained some coverage of sample points. Sediment points in the Eastern Bering Sea are separated on average by ˜10.5 km, while some localized sampling efforts, especially near shore, were at a much greater density. Very few point were located outside of the Continental Shelf.

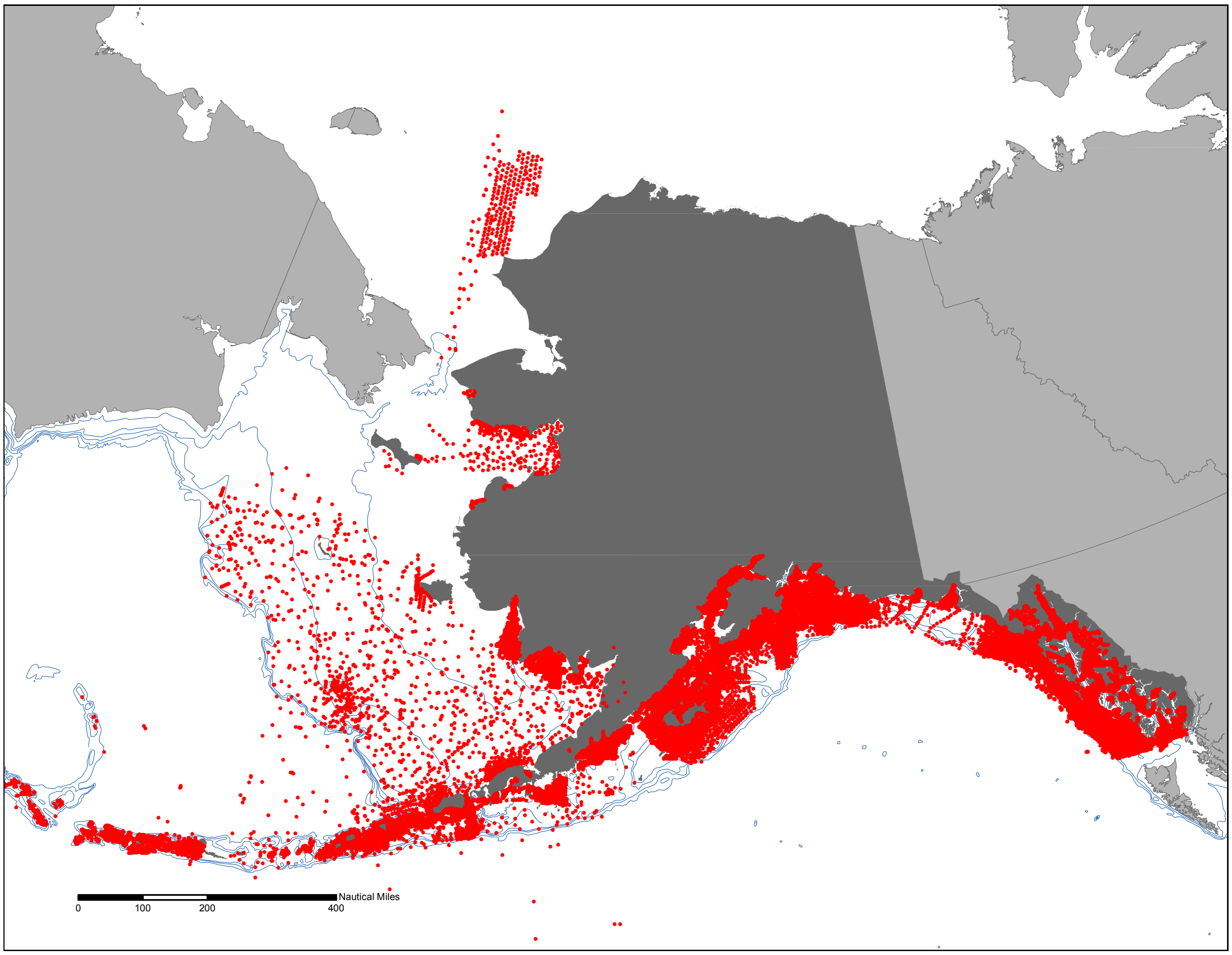

Figure 2: Sediment sample locations.

Sediment sample locations.

We first organized the sediment descriptions into 15 common terms that corresponded roughly to sediment classes in the FE model (Table 8). We then mapped the 15 terms to the five sediment types:

1. mud ← mudsandgravel, mudsand, mudgravel, mud, soft, silt, clay
2. sand ← mudsandgravel, sand, mudsand, coarse, soft
3. granule-pebble ← mudsandgravel, mudgravel, gravel,pebble,coarse
4. cobble ← cobble,rock, hard
5. boulder ← boulder

The mapping was not one-to-one, such that more than one sediment class could be present at a sampled point. Consequently, we treated each sediment class as a binary presence/absence variable at each point. Using all sediment points, we used an indicator Kriging approach (Geostatistical Wizard, ArcMap v10.2) to interpolate probability surfaces for each sediment class. We set a threshold at 0.5 to indicated presence/absence and estimated it to a 2.5 km grid aligned with the 5 km grid used in the FE model (Fig. XXXX). Thus, four sediment grid cells were located within each 5 km grid cell, providing a pseudo-area weighted measured of each sediment type within each 5 km grid cell. For each 5 km grid cell, the proportion of each sediment type was calculated as the sum of all 2.5 km grid cells with sediment present (up to four for each sediment class) divided by the sum of all present cells across all sediments (up to 20 possible, 4 cells X 5 sediment classes). In ~10% of the 5 km grid cells, no sediment class was predicted present. In theses cases, we used the sediment proportions from the nearest 5 km grid cell.

4 Model implementation

4.1 Model implementation overview

Implementation of the FE model is split between two processes. The first process contains most on the GIS analysis and is written in Python (v. 2.7.5). Its main product is a table of fishing effort by grid cell, month, and gear type. The second process uses the fishing effort table, along with, recovery and susceptibility tables to track undisturbed habitat H through time. The second process is written in R (3.1.2). Fig. 3 shows a schematic of both processes with their inputs and outputs.

The Python script relies heavily on the *arcpy* module from ArcMap (v. 10.2.1) to run the GIS tools. Because of the dependency on the *arcpy* module, a valid ArcGIS license

Table 8: Sediment description

Class	Term	Description
boulder	boulder	bld, blds, boulder, boulders
mud	clay	c, cl, cla<, clay
sand, granule-pebble	coarse	coarse, coasrse, corse, crs, grs, gs, sg, [g]s
cobble	cobble	co, cobble, cobbles
granule-pebble	gravel	g, gr, grave, gravel, gravels, gravll
cobble	hard	hard, hrd
mud	mud	coarsemud, m, md, mud, muddy, muds, mug, muid
mud, granule-pebble	mudgravel	[g]m, gm, mg
mud, sand	mudsand	ms, sm
mud, sand, gravel	mudsandgravel	[g]ms, [g]sm, gms, msg
granule-pebble	pebble	p, pbls, peb, pebbles, pebble, pebles, pebs, stone, stones
cobble	rock	*rk, *rock, r, r*, rck, rcy, rk, rk*, rks, rky, rkysh, rock, rocks, rocky
sand	sand	s, sand, sandy, sanf, sd, sdy
mud	silt	silt, silty
mud, sand	soft	sft, soft

is required to run the script. Three input files are required to run the script:

1. CIA dataset: Geodatabase polyline feature class of historic tows.
2. Grid5k: 5 km X 5 km grid across North Pacific used for spatial units of analysis.
3. Gear buffers: Geodatabase table of buffer size for each gear type (Table 7).

The Python script will produce three output files, the first two of which are intermediate, and the last to be used as input into the R script:

1. CIA_data_buffered: Goedatabase polygon feature class of the CIA dataset buffered by values from gear buffers .
2. Buffers_Grid5k_intesect: Goedatabase polygon feature class of the spatial intersection of the buffered CIA dataset and Grid5k.

3. aggreagted_fishing_effort.dbf: DBASE table of fishing effort, f, by grid cell, year, month, and gear type.

The second process, written in R requires four inputs, the first being the output table from the first process:

1. aggreagted_fishing_effort.dbf: DBASE table output from Python script.
2. Susceptibility tables: CSV tables of susceptibility for habitat feature-sediment class-gear type combinations. Three tables total, one for each gear type.
3. Recovery table: CSV table of recovery for each habitat feature-sediment class combination.
4. Sediment proportions: DBASE table of sediment class proportions for each grid cell.

The R script produces a single output:

1. FE_model_output: CSV table of total undisturbed habitat for each grid cell and month.

4.2 Python script detail

Step 1. Import modules, set *arcpy* environment settings and define data input and outputs.

```
import os, time, arcpy
from datetime import datetime

arcpy.env.workspace = "WORKING DIRECTORY"
arcpy.env.overwriteOutput = True
arcpy.env.qualifiedFieldNames = False

gdb = "Fishing_effects_model.gdb"

## Feature class import
fishingEffort = os.path.join(gdb, "VMS_OBS_UnOBS_wVessel_Singlepart")
```

```
Grid5k = os.path.join(gdb, "Grid5k")



### Feature class products
fishingEffort_buffers = os.path.join(gdb, "VMS_OBS_UnOBS_wVessel_Buffer")

Grid_FE_intersect = os.path.join(gdb, "Grid_FE_intersect")

fldList2Transfer = "YEAR;MONTH;BufferCode"  #Fields to keep from
                                             #fishingEffort_buffers



## Tables to import
bufferSize = os.path.join(gdb, "FishingGear_buffers")



## Output tables
aggregatedFishingEffort = "aggregated_fishing_effort.dbf"
```

Step 2. Define buffer function. Each tow in the the CIA dataset buffered based on the fishery.

```
############## Buffer function ################################
def FishingEffort_buffer(inFeatures, joinFields, joinTable, outBuffer, bufferField):

    print "Joining bufferSize to Fishing Effort lines...."
    arcpy.MakeFeatureLayer_management (inFeatures,  "inLayer")

    # Join the feature layer to a table
    arcpy.AddJoin_management("inLayer", joinFields, joinTable, joinFields)

    print "Starting Fishing Effort buffers..."
    bufferStart = time.time()
    arcpy.Buffer_analysis("inLayer", outBuffer, bufferField, "FULL", "FLAT", "NONE")
    bufferEnd = time.time()
    print "Buffer completed in %d minutes" % int((bufferEnd-bufferStart)/60)
```

Step 3. Define intersect function. All buffered tows will be intersected with the 5 km Grid to get a nominal area swept for individual tows in grid cells. We do not use the standard *Intersect* tool here, as it results in unnecessarily long processing time given the size of the CIA dataset and the number of grid cells. Instead we iterate over all features in both input feature classes and use intersect geometry methods to intersect features one pair at a time. This function was adapted from the *PairWiseIntersect.py* tool (Hartling, 2013)

```
########## FastIntersect function ################################
def FastIntersect(inputFC1, inputFC2, outputFC, fldList2Transfer):
  # Prep for processing
  # Determine if the inputs are layers or featureclass
  inputLayer1 = arcpy.MakeFeatureLayer_management(inputFC1,"inputLayer1")

  inputLayer2 = arcpy.MakeFeatureLayer_management(inputFC2,"inputLayer2")

  layer1Type = arcpy.Describe(inputLayer1).shapeType
  layer2Type = arcpy.Describe(inputLayer2).shapeType

  # Get the input geometry type for use in the geometry intersect method
  dimension = 4

  arcpy.AddMessage(time.ctime())
  startProcessing = time.time()

  # Setup input fields
  tempFldsInput1 = [f.name.upper() for f in arcpy.ListFields(inputLayer1)]
  fldsInput1 = list(tempFldsInput1)
  fldsInput1.remove(arcpy.Describe(inputLayer1).shapeFieldName.upper())
  fldsInput1.remove(arcpy.Describe(inputLayer1).oidFieldName.upper())
  fldsInput1.append("shape@")

  fldsInput2Orig = arcpy.ListFields(inputLayer2)
  fldsInput2 = fldList2Transfer.upper().split(";")


  fldsInput2.append("shape@")

  # Setup the output feature class for receiving spatial data from the
```

```
# intersect operation and attribute data from both the inputs.
arcpy.CreateFeatureclass_management(os.path.dirname(outputFC),
                                    os.path.basename(outputFC),
                                    layer1Type,
                                    inputLayer1,
                                    spatial_reference=inputLayer1)
arcpy.MakeFeatureLayer_management(outputFC, r"outputLayer")

fldsInput2Modified = []
for fld in fldsInput2:
    if fld == "shape@":
      pass
    else:
      for fldOrig in fldsInput2Orig:
        if fldOrig.name.upper() == fld:
          if fld in fldsInput1:
            newFld = fld + "_1"
            fldsInput2Modified.append(newFld)
          else:
            newFld = fld
          arcpy.AddField_management(r"outputLayer", newFld, fldOrig.type)
          break

tempFldsOutput = [f.name.upper() for f in arcpy.ListFields(r"outputLayer")]
fldsOutput = list(tempFldsOutput)
fldsOutput.remove(arcpy.Describe(r"outputLayer").shapeFieldName.upper())
fldsOutput.remove(arcpy.Describe(r"outputLayer").oidFieldName.upper())
fldsOutput.append("shape@")

# Make sure to only process features in input1 that intersect
# something in input2.
print "Selecting Grid cells that contain Fishing effort..."
arcpy.SelectLayerByLocation_management(inputLayer1, "INTERSECT", inputLayer2)


# Count number of feature in inputLayer1 so we can keep track of
# progress in 5% increments
arcpy.MakeTableView_management(inputLayer1, "myTableView")
```

```
nFeatures = int(arcpy.GetCount_management("myTableView").getOutput(0))
nFeatures_5pct = int(nFeatures/20)

# Intersect each input feature with the features from the second
# input feature class and determine the field values to be transfered
# to the output
print "Processing features..."
intersectStart = time.time()
inCursor = arcpy.da.InsertCursor(r"outputLayer", fldsOutput)
with arcpy.da.SearchCursor(inputLayer1, fldsInput1) as cursor:
  for cnter, row in enumerate(cursor, 1):
      if cnter%500 == 0:
        print "%d features processed of %d features..." %(cnter, nFeatures)
      arcpy.SelectLayerByLocation_management(inputLayer2, "INTERSECT", row[-1])
      with arcpy.da.SearchCursor(inputLayer2, fldsInput2) as cursor2:
        for row2 in cursor2:
          clippedFeature = row2[-1].intersect(row[-1], dimension)
          # Determine the field values to insert in the output
          flds2Insert = list(fldsOutput)
          for i, outFlds in enumerate(fldsOutput):
            found = False
            # Process the first layers attribute values
            for j, input1Fld in enumerate(fldsInput1):
              if input1Fld != "shape@" and outFlds != "shape@":
                if outFlds == input1Fld:
                  flds2Insert[i] = row[j]
                  found = True
                  break
            # Process the second layers attribute values
            if found == False:
              for f, fldIn2 in enumerate(fldsInput2):
                if fldIn2 != "shape@" and outFlds != "shape@":
                  if outFlds in fldsInput2 or outFlds in fldsInput2Modified:
                    if fldIn2 == outFlds or fldIn2 + "_1" == outFlds:
                      flds2Insert[i] = row2[f]
                      break
                    else:
                      break
```

```
                    else:
                      break
            flds2Insert[-1] = clippedFeature
            inCursor.insertRow(flds2Insert)
  try:
    del inCursor
    del cursor, cursor2
  except:
    pass
  intersectEnd = time.time()

  print "Intersect completed in %d minutes" % int((intersectEnd-intersectStart)/60)
```

Step 4. Define function to parse month. We extract month of tow from "M/D/Y" format.

```
def parseMonth(inputFC, dateField, monthField):
    arcpy.MakeTableView_management(inputFC, "myTableView")
    nFeatures = int(arcpy.GetCount_management("myTableView").getOutput(0))

    print "Adding month field...."
    arcpy.AddField_management(inputFC, monthField, "TEXT", "", "", 2)

    print "Starting month parsing...."
    rows = arcpy.UpdateCursor(inputFC)

    for cnter, row in enumerate(rows, 1):
        if cnter%250 == 0:
            print "%d features processed of %d features..." %(cnter, nFeatures)

        datetimeVal = row.getValue(dateField)
        formattedTime = datetime.strftime(datetimeVal, "%m/%d/%Y")
        month = formattedTime.split("/")[0]
        row.setValue(monthField, month)
        rows.updateRow(row)

    del rows
```

Step 5. Run the the functions defined above.

```
## Run buffer function
FishingEffort_buffer(inFeatures = fishingEffort,
                     joinFields ="BufferCode",
                     joinTable = bufferSize,
                     outBuffer = fishingEffort_buffers,
                     bufferField = "FishingGear_buffers.BufferSize")
                     # BufferSize has only been temporarily joined so we need
                     # to identify field with original layer



## Run month parser
parseMonth(inputFC = fishingEffort_buffers,
           dateField = "WEEK_END_D",
           monthField = "MONTH")


## Run intersect
FastIntersect(inputFC1 = Grid5k,
              inputFC2 = fishingEffort_buffers,
              outputFC = Grid_FE_intersect,
              fldList2Transfer = fldList2Transfer )
```

Step 6. Aggregate nominal area swept for each grid cell and gear type.

```
## Run aggregate
print "Aggregating fishing effects..."
arcpy.Statistics_analysis(Grid_FE_intersect, aggregatedFishingEffort,
                          [["Shape_Area", "SUM"]],
                          ["Grid5k_ID","YEAR","MONTH","AGENCY_GEA"])
```

4.3 R script detail

The R code to trach H through time is detailed below. Like the Python code, all necessary components are included in the shaded text. The shaded code blocks need to be run in sequence for the model to work properly. Two external packages are required, *foreign* and *rehsape2*, for the model to work. These both are available through the Comprehensive R Archive Network (CRAN; http://cran.r-project.org/).

5 Fishing effects discrete time model

Step 1. Load packages, import data, and define variables.

```
library(foreign)
library(reshape2)
library(rgdal)
library(maptools)

# Set working directory
setwd("C:\\Users\\T.Scott\\Dropbox (JBER-APU)\\NOAA_GIS_data\\scott_products")




# Import data
grid5k = readOGR(dsn = "Fishing_effects_model.gdb", layer = "Grid5k")
grid5k.dat = grid5k@data

fe = read.dbf("aggregated_fishing_effort.dbf")
fe_sasi = read.csv("susceptibiltiy matrices\\FE_SASI_gear_link.csv")





# Set variables
nYears = length(unique(fe$YEAR))
nSubAnnual = 12
nGrid = length(unique(fe$Grid5k_ID))
nGear = length(unique(fe$BUFFERCODE))
nSubst = 5



grid_order = sort(unique(fe$Grid5k_ID))
```

```
subst_types = c("Mud", "Sand", "Gran.Peb", "Cobble", "Boulder")

SASI_gears = c("trawl", "longline", "trap")

gear_types = levels(fe$BUFFERCODE)

eg = expand.grid(Grid5k_ID=unique(fe$Grid5k_ID), Gear=unique(fe$BUFFERCODE))

m = merge(grid5k.dat, fe, by = "Grid5k_ID")

m$prop = m$SUM_AdjAre/m$Shape_Area

m$MONTH = as.numeric(as.character(m$MONTH))
```

Step 2. Convert long form data calculated in the Python script (`aggregated_fishing_data.dbf`) to a four dimensional array (`F_a`). The first two dimensions of the array index over years and months. The third and fourth dimensions hold a matrix of fishing effort, f, by grid cell and gear type.

```
# Populate Fishing effort array
F_a = array(NA, dim = c(nYears, nSubAnnual, nGrid, nGear))

year.i = 1  # year counter

for(year in min(m$YEAR):max(m$YEAR)){
  my = subset(m, YEAR == year)
  for(month.i in 1:12){
    mym = subset(my, MONTH==month.i)

    mym = merge(x = eg, y = mym,
                by.x = c("Grid5k_ID","Gear"),
                by.y = c("Grid5k_ID", "BUFFERCODE"),
                all.x=T)

    mym[is.na(mym$prop),]$prop = 0

    mym.x = dcast(Grid5k_ID ~ Gear, data=mym, value.var = "prop")
```

```
    mym.x = mym.x[order(mym.x$Grid5k_ID),]
    mym.x = mym.x[,c(gear_types)]

    F_a[year.i, month.i, ,] = as.matrix(mym.x)



  }
  year.i = year.i + 1
}
```

Step 3. Create random susceptibilities based on suceptibility class.

```
# Suceptibility
gear.q = matrix(NA, nrow = 3, ncol = nSubst)

i = 1
for(gear in SASI_gears){
  gear.m = read.csv(paste("R_input_tables\\Susceptibilty_table_",
                          gear, ".csv", sep=""))

  gear.m = gear.m[,subst_types]

  for(column in 1:ncol(gear.m)){
    gear.m[gear.m[,column] %in% 0, column] =
      runif(sum(gear.m[,column] %in% 0), min = 0, max = 0.1)

    gear.m[gear.m[,column] %in% 1, column] =
      runif(sum(gear.m[,column] %in% 1), min = 0.1, max = 0.25)

    gear.m[gear.m[,column] %in% 2, column] =
      runif(sum(gear.m[,column] %in% 2), min = 0.25, max = 0.5)

    gear.m[gear.m[,column] %in% 3, column] =
      runif(sum(gear.m[,column] %in% 3), min = 0.5, max = 1)
  }

gear.q[i,] = colMeans(gear.m, na.rm=T)
```

```
i = i + 1

}

gear.q.df = data.frame(SASI_gear = SASI_gears, gear.q)
names(gear.q.df)[-1] = subst_types


fe_sasi = merge(fe_sasi, gear.q.df, by = "SASI_gear", all=T)

fe_sasi = fe_sasi[match(gear_types, fe_sasi$FE_gear),]

q_m = as.matrix(fe_sasi[,subst_types])
```

Step 4. Populate impact array, `I.prime\_a`, a four-dimensionl array of fishing impact, I'. The first two dimensions index over years and months, the last two dimensions hold a grid cell by subtrate class matrix of I'. We first calculate I using Eq. 6 then convert to I' using Eq. 4.

```
#Fishing impacts (I')

I_a = array(NA, dim = c(nYears, nSubAnnual, nGrid, nSubst))

for(y in 1:nYears){
  for(m in 1:nSubAnnual){
    I_m = F_a[y,m,,] %*% q_m
    I_a[y,m,,] = 1-exp(-I_m)
    }
  }
```

Step 4. Populate recovery array, `rho\_a`, a four dimensional array of recovery, ρ'. The first two dimensions index over years and months, the last two dimensions hold a grid cell by subtrate class matrix of ρ'. We first create random recovery times (τ) based on the recovery classification for each habitat feature-sediment class then calculate the mean recovery time each sediment class. Recovery times are converted to recovery rates, ρ We then populate `rho.prime_a` assuming constant recovery across year, months, and grid cells. We use Eq. 8 to convert ρ to ρ', dividing by twelve (`nSubAnnual`) to convert recovery time in years to months. This assumes a constant rate of recovery within a year.

```
# Recovery (rho')
tau_m = read.csv("R_input_tables\\Recovery_table.csv")

tau_m = tau_m[,subst_types] # Make sure sediments are in correct order



for(column in 1:ncol(tau_m)){
  tau_m[tau_m[,column] %in% 0, column] =
    runif(sum(tau_m[,column] %in% 0), min = 0, max = 1)

  tau_m[tau_m[,column] %in% 1, column] =
    runif(sum(tau_m[,column] %in% 1), min = 1, max = 2)

  tau_m[tau_m[,column] %in% 2, column] =
    runif(sum(tau_m[,column] %in% 2), min = 2, max = 5)

  tau_m[tau_m[,column] %in% 3, column] =
    runif(sum(tau_m[,column] %in% 3), min = 5, max = 10)
}

tau_v = colMeans(rho_m, na.rm=T)

rho_v = 1 / (tau_v * nSubAnnual)  # Convert recovery time in years to rates per month

rho.prime_a = array(NA, dim = c(nYears, nSubAnnual, nGrid, nSubst))

for(y in 1:nYears){
  for(m in 1:nSubAnnual){
      rho.prime_a[y,m,,] = 1-exp(-rho_v)

    }
  }
```

Step 5. Import sediment data.

```r
# Sediment
sed = read.dbf("sediment_table\\sediment_v4.dbf")

# Create sediment matrix for model.  Make sure grid order is same as I_a
# and keep only sediment areas
sedProps = as.matrix(sed[match(grid_order, sed$Grid5k_ID),
                  c("mudProp","sandProp","grpeProp","cobProp","bouldProp") ])
```

Step 6. Define fishing effects model function. This function tracks undisturbed habitat, H, across grid cell, years, and month, and sediment type. Inputs are the impact array, `I_a`, recovery array, `rho_a`, intitial habitat conditions, `H_prop_0`, and sediment proportions, `sedProps`. The output is a four dimensional array of proportion of undisturbed habitat indexed over years, months, grid cells, and sediment classes. These proportions do not account for the amout of each sediment class within a grid cell. For example, this function will calculate a value for a sediment class that does not occur within the grid cell. Thus, the ouput from this function is not meaningful until scaled by the sediment class proprtions (see Step 9).

```r
# Fishing Effects Model
FishingEffectsModel = function(I.prime_a, rho.prime_a, H_prop_0){
  model_nYears = dim(I.prime_a)[1]
  model_nSubAnnual = dim(I.prime_a)[2]
  model_nGrid = dim(I.prime_a)[3]
  model_nSubst = dim(I.prime_a)[4]

  #Make array to hold H
  H_prop = array(dim = c(model_nYears, model_nSubAnnual, model_nGrid, model_nSubst))

  for(y in 1:model_nYears){
    for(m in 1:model_nSubAnnual){

      if(y == 1 & m == 1){      # First time step use H_prop_0 for t-1
        prior_state = H_prop_0
        } else if (m == 1){
          prior_state = H_prop[y-1,model_nSubAnnual,,]
          } else{
            prior_state = H_prop[y,m-1,,]
```

```
          }

      H_from_H = (1-I.prime_a[y,m,,])*prior_state  # undisturbed remaining undisturbed
      H_from_h = (1-prior_state) * (rho.prime_a[y,m,,]) # disturbed recovered to undisturbed
      H_prop[y,m,,] = H_from_H + H_from_h  # Total proportion disturbed

      }
    }

  return(H_prop)
}  # end function
```

Step 7. Defining intial conditions. `H_prop_0` is a matrix of intial H values over grid cells and sediments classes. Like the output from fishing effect model function (Step 6), these proportions are independent of how much of each sediment there is in a grid cell. Thus, we can use output from Step 6 to calculate `H_prop0`. Here, we use a starting condition of 100% undisturbed habiat across all grid cells and sediment classes. The commented out code shows an example of how we might use the first five years of fishing data to "burn-in" sensible intial conditions. Note commented code still requires the inital definition of `H_prop_0`. See Section 7.1 for a discussion of the sensitivity of the model to the intial conditions.

```
# Define initial conditions
H_prop_0 = matrix(1, nrow = nGrid, ncol = nSubst)

## Example of five year burn-in
# First run the model for the first five years
# H_burn = FishingEffectsModel(I.prime_a[1:5,,,], rho.prime_a[1:5,,,], H_prop_0)

#Then extract only the last time step as the new intial condition
# H_prop_0 = H_burn[5,12,,]
```

Step 8. Run the fishing effects model function.

```
H_tot = FishingEffectsModel(I.prime_a, rho.prime_a, H_prop_0)
```

Step 9. Create GIS file of disturbed habitat by grid cell across time.

```r
# Calulate undisturbed Areas
tracked_cells = grid5k.dat[match(grid_order, grid5k.dat$Grid5k_ID), ]


undistAreas = matrix(NA, ncol = nSubAnnual*nYears, nrow = length(grid_order))

i = 1
for(y in 1:nYears){
  for(m in 1:12){
    undistAreas[,i] =
      rowSums(H_tot[y,m,,]*sedProps*tracked_cells$Shape_Area/1000000)

    i = i + 1
    }
}


undistAreas = data.frame(Grid5k_ID = grid_order, undistAreas)

months = c("Jan","Feb","Mar","Apr", "May", "Jun", "Jul","Aug",
           "Sep","Oct","Nov","Dec")
year = 2003

for(i in 1:(ncol(undistAreas)-1)){
  month = i %% 12
  if(month == 0) month = 12
  names(undistAreas)[i+1] = paste(months[month], year, sep="")
  if(month == 12) year = year + 1
}


disturbAreas = data.frame(Grid5k_ID = undistAreas[,1],
                          apply(undistAreas[,-1],2,
                                function(x) tracked_cells$Shape_Area/1000000 - x))

# If disturbance is < 1 sq. m turn to zero
disturbAreas[abs(disturbAreas) < 1e-6] = 0
```

```r
grid5k@data = merge(grid5k@data, disturbAreas, by = "Grid5k_ID", all.x = T)

# Turn NAs to zero.  These are places that were never fished
grid5k@data[is.na(grid5k@data)] = 0


writeSpatialShape(grid5k, "FE_model_output\\Grid5k_fishing_effects.shp")
```

6 Comparison to LEI

We developed the FE model to behave similarly to LEI (Fujioka, 2006; Sethi et al., 2014). The FE model, like LEI, defines two habitat states, H and h, and models their dynamics through fishing impacts, I and recovery, ρ dynamics. The primary difference between the two model is that transitions between H and h is a continuous process in LEI, but is modeled as a discrete time process in the FE model.

LEI is based on a set of differential equations, (Eq. 1 and 2 in Fujioka 2006),

$$\mathrm{d}H/\mathrm{d}H = -IH + \rho h \tag{17}$$

$$\mathrm{d}h/\mathrm{d}H = +IH - \rho h \tag{18}$$

Solving these equations for H_t (Eq. 3 in Fujioka 2006),

$$H_t = H_0^*[I\mathrm{e}^{-(I+\rho)t} + \rho]/(I + \rho) \tag{19}$$

We added the * to H_0 to distinguish it from H_0 in the FE model, as it has a different definition. H_0^* is the total amount of habitat that can be impacted by fishing. For example, if we assume all habitat can be impacted, $H_0^* = 1$. We can evaluate the long term equilibrium H_{eq}, finding the limit of Eq. 19 as $t \to \infty$ (Eq. 4 in Fujioka 2006),

$$H_{eq} = H_0^*\rho/(I + \rho) \tag{20}$$

We compared H_{eq} to the long term behavior of the FE model (Fig. 4). We dropped the H_0^* as we assumed all habitat (100%) is able to be impacted, and thus $H_0^* = 1$. We considered 16 combinations of I and ρ to compare the two models. For the FE model, we assumed I and ρ remained constant. Also, for the sake of this analysis, we can assumed both I and ρ are rates given in the same arbitrary time increment, thus we did not divide by 12 or use any other time conversion.

We can see from Fig. 4 that when $\rho = I$ (plots along the diagonal), H_{eq} is equivalent to the long term stable equilibrium of the FE model. When $\rho > I$ (plots above the diagonal) H_{eq} is greater than the long term stable equilibrium of the FE model. And, lastly, when $\rho < I$ (plots below the diagonal), H_{eq} is less than the long term stable equilibrium of the FE model. Because the impact and recovery processes can only act on the undisturbed or disturbed habitat from the previous time step, any habitat disturbed by fishing impacts in the current time step will not have the chance to recover until subsequent time steps. Similarly, habitat that recover in the current time step cannot but impacted by fishing until subsequent time steps. Thus, there is a one time step lag between these two processes. Because LEI is modeled on a continuous time framework, it does not have the same inherent time lag. As a consequence, when recovery out paces impact ($\rho > I$), the long term behavior of the time discrete model shows *more* reduction of habitat than LEI. This is due to the recovery process not being able to continuously function on disturbed habitat. The impact process obviously lags as well, but since the impact rate is less than recovery, this results in a reduction in overall recovery compared to LEI. The same mechanism holds true but in reverse when the impact rate is greater than recovery.

We can identify similar behavior analytically. We first simplify Eq. 2 terms of only H using Eq. 1. We have removed the t subscript from I' and ρ' because we are assuming constant impact and recovery over time,

$$H_{t+1} = H_t - I'H_t + \rho' h = H_t(1 - I' - \rho') + \rho'$$

Let $\quad u = 1 - I' - \rho'$

$$\textbf{Time 1} \quad H_1 = H_0 u + \rho'$$

$$\begin{aligned} \textbf{Time 2} \quad H_2 &= H_1 u + \rho' \\ &= [H_0 u + \rho']u + \rho' \\ &= H_0 u^2 + \rho' u + \rho' \end{aligned}$$

$$\begin{aligned} \textbf{Time 3} \quad H_3 &= H_2 u + \rho' \\ &= [H_0 u^2 + \rho' u + \rho']u + \rho' \\ &= H_0 u^3 + \rho' u^2 + \rho' u + \rho' \\ &= H_0 u^3 + \rho'(u^2 + u + 1) \end{aligned}$$

After three time steps we can begin to see the pattern to the recursion. Generalizing H_t in term of H_0,

$$H_t = H_0 u^t + \rho'(u^{t-1} + u^{t-2} \ldots u^1 + u^0) = H_0 u^t + \rho' \sum_0^t u^t \tag{21}$$

To compare the FE model to LEI, we are interested in the long term stable value of $H_{t\to\infty}$. Because both I' and ρ' are bounded on the interval (0, 1), u is bounded by (-1, 1). Thus, as $H_{t\to\infty}$, the first term in Eq. 21, $H_0 u^t \to 0$. The second term, $\rho' \sum_0^t u^t$ is composed of ρ' times a power series. The solution of a power series is,

$$f(u) = \sum_{k=0}^{\infty} u^k = \frac{1}{1-u} \qquad \text{for } |u| < 1 \tag{22}$$

Thus,

$$H_{t\to\infty} = \rho' \frac{1}{1-u} = \frac{\rho'}{\rho' + I'} \tag{23}$$

$$H_{t\to\infty} = \frac{\mathrm{e}^{2\rho+I} - \mathrm{e}^{\rho+I}}{2\mathrm{e}^{2\rho+I} - \mathrm{e}^{2\rho} - \mathrm{e}^{\rho+I}} \tag{24}$$

$$H_{t\to\infty} = \frac{\mathrm{e}^{(2\rho+I)\Delta t} - \mathrm{e}^{(\rho+I)\Delta t}}{2\mathrm{e}^{(2\rho+I)\Delta t} - \mathrm{e}^{2\rho\Delta t} - \mathrm{e}^{(\rho+I)\Delta t}} \tag{25}$$

7 Preliminary Results

7.1 Sensitivity of initial conditions

References

Fujioka, J. T. (2006). A model for evaluating fishing impacts on habitat and comparing fishing closure strategies. *Canadian Journal of Fisheries and Aquatic Sciences*, *63*(October), 2330–2342. doi:10.1139/f06-120

Hartling, K. (2013). PairWiseIntersect.py. ESRI. Retrieved from http://blogs.esri.com/esri/arcgis/2013/05/20/are-you-sure-intersect-is-the-right-tool-for-the-job/

New England Fishery Management Council (NEFMC). (2011). *The Swept Area Seabed Impact (SASI) approach: a tool for analyzing the effects of fishing on essential fish habitat.* New England Fishery Management Council Report. Newburyport, MA.

Sethi, S., Harris, B. P., & Rose, C. S. (2014). *Examination of the Fujioka fishing effects model: model formulation , implementation, and interpretation.* The Fishersies, Aquatic Science & Technology (FAST) Laboratory, Alaska Pacific University. Anchorage, AK.

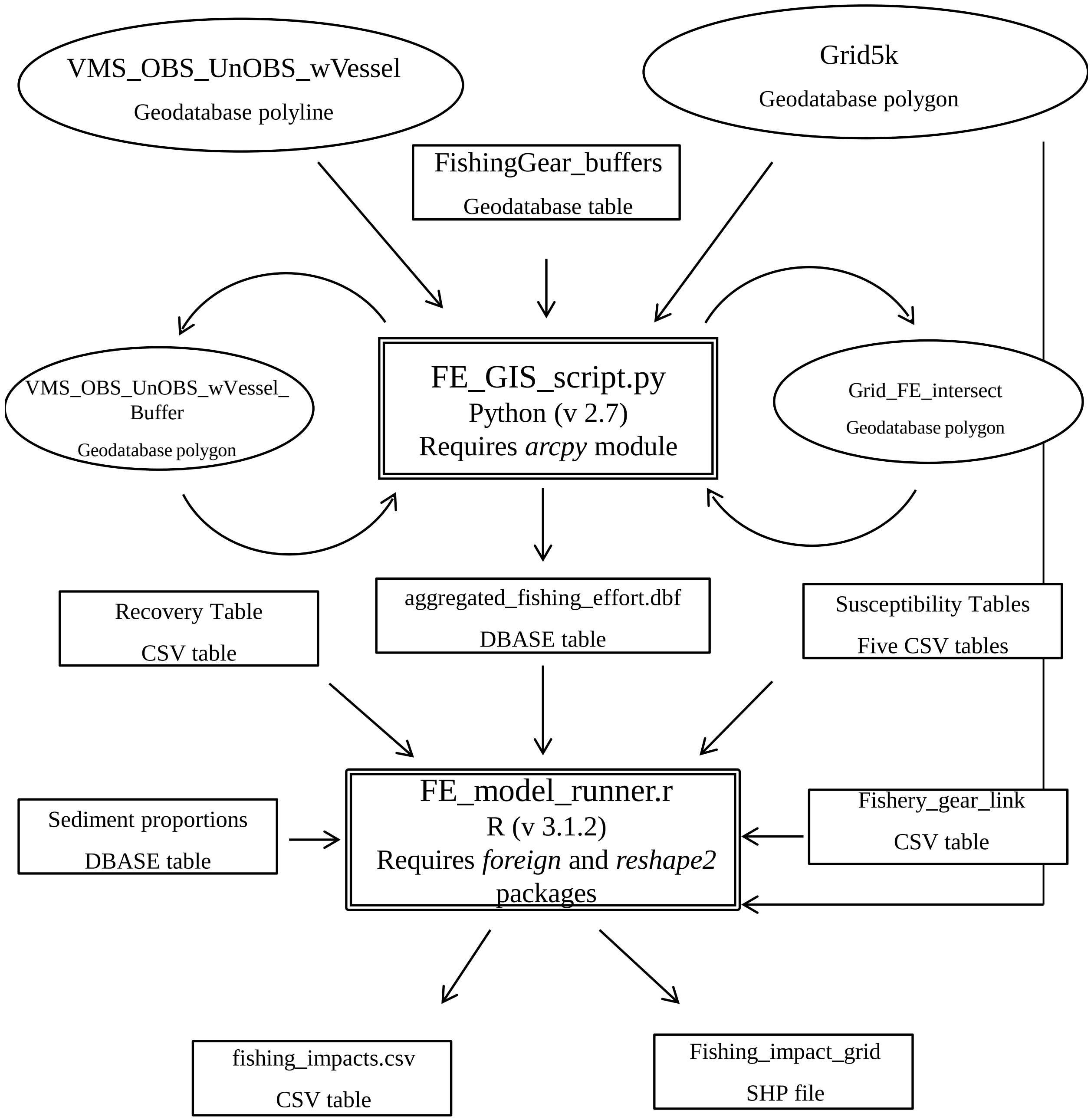

Figure 3: Schematic of FE model implementation.

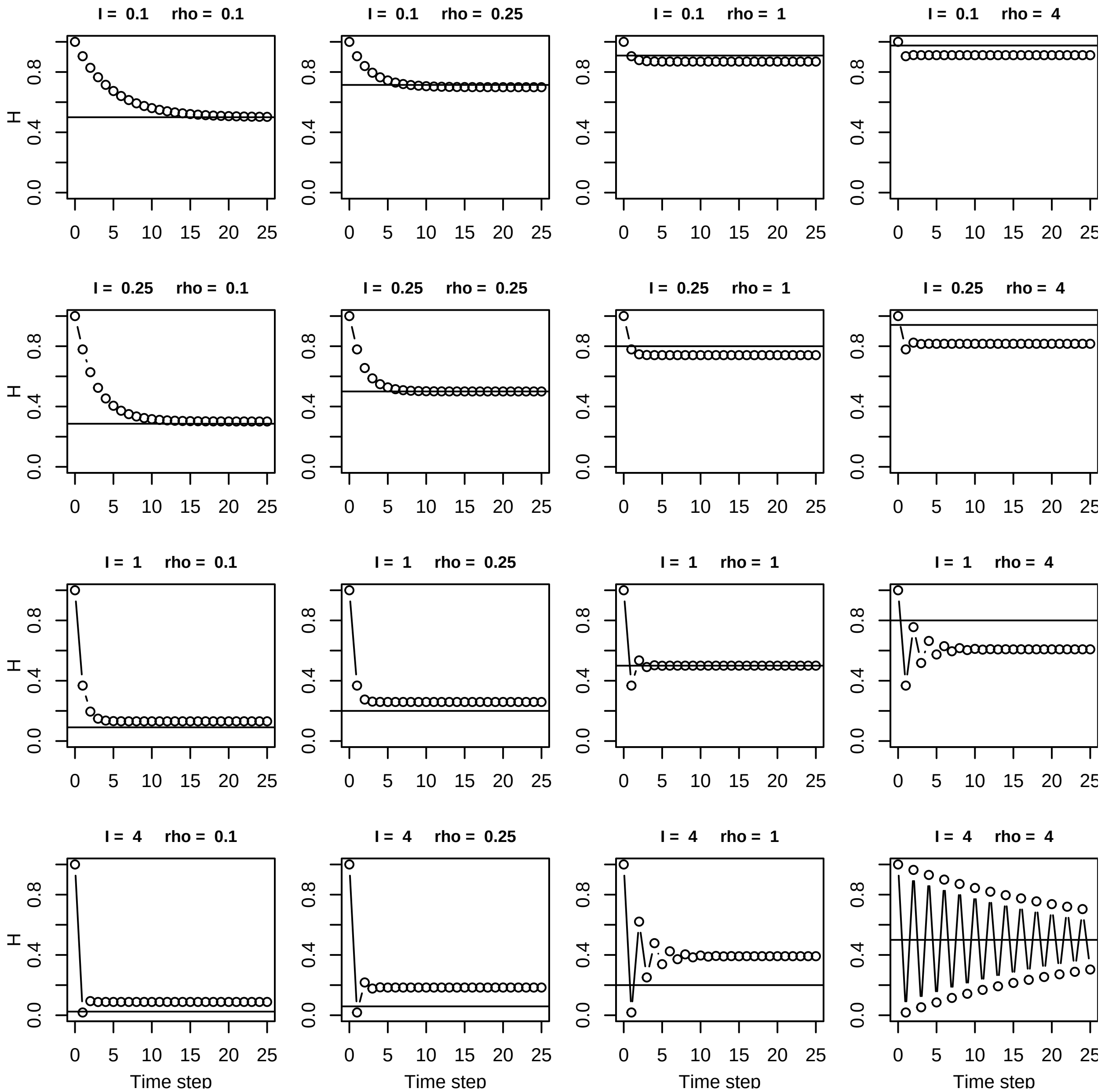

Figure 4: Comparison of FE long term behavior to LEI.
The FE model was run over 25 time steps at various I and ρ values. Initial conditions, H_0 was set to one. Black horizontal lines show H_{eq} (Eq. 20) estimates using the same I and ρ.