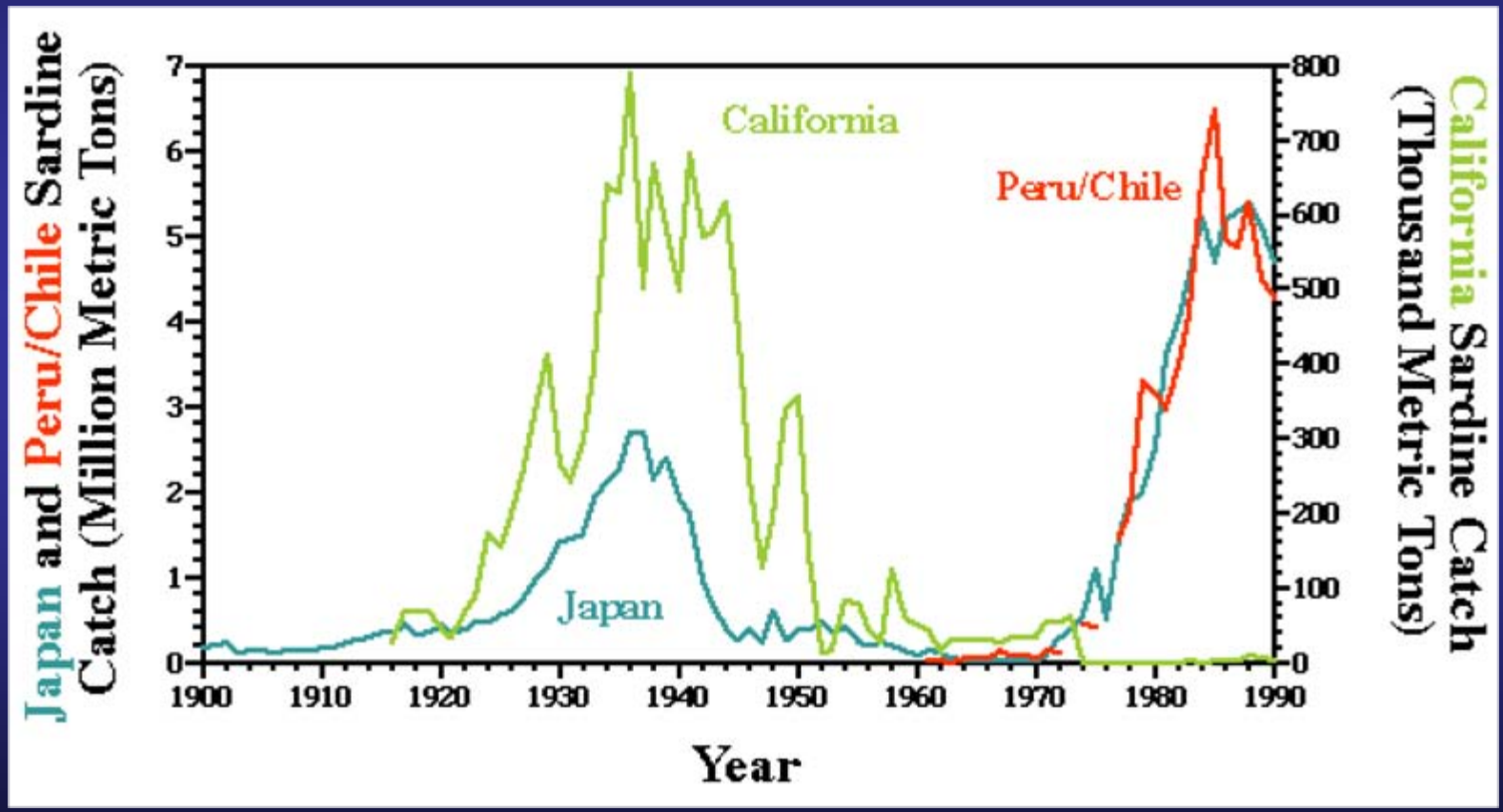
Development of a climate-to-fish-to-fishers model: data structures and domain decomposition

Kate Hedstrom,
Jerome Fiechter, Kenny Rose,
Enrique Curchitser, Miguel Bernal,
Shin-ichi Ito, Bern Megrey

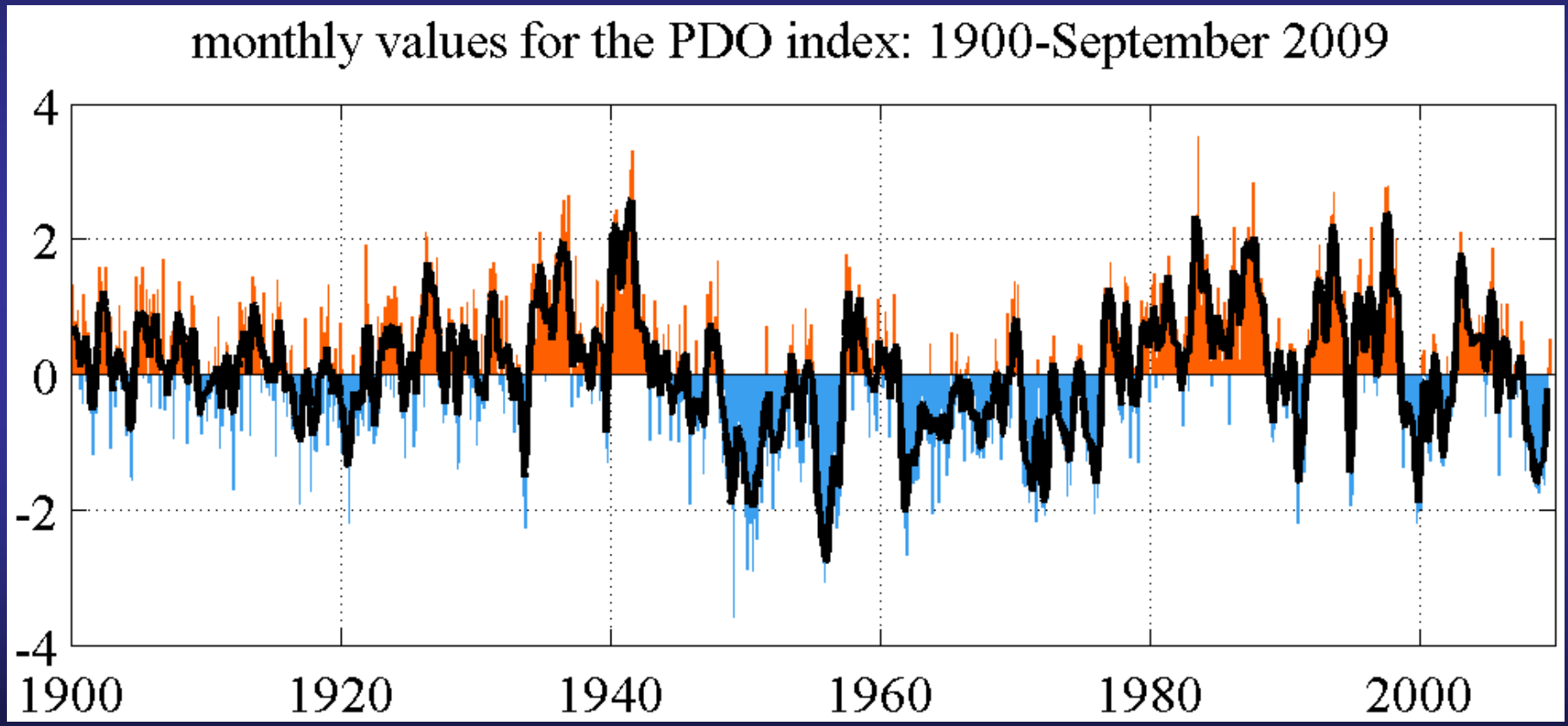
Outline

- Goals
- Methods
- Conclusions and future work

Sardine Landings in the Pacific



Pacific Decadal Oscillation



From Nate Mantua

Goals

- These cycles in small pelagic fish go back thousands of years: we'd like to build a model that can reproduce these cycles
- Couple to physical model that can reproduce the PDO (ROMS)
- Start with fish and add fishing fleets later

Sardines and Anchovies

- Both have periods of abundance and periods of scarcity
- Both have life histories that differ between warm and cold years:
  - Age and size of spawning fish
  - Age of oldest fish
  - Migration pattern and location of spawning

Superindividual Models

- Lagrangian tracking of fish
- Each “fish” represents a group, not just one individual
- The superindividual starts with a large “worth” as eggs, then worth is reduced as individuals die
- Include the “big four” – spawning, growth, behavior, mortality

Many Challenges

- Behavior should include spawning and feeding migrations, predator avoidance, etc.
- Growth requires knowledge of bioenergetics – grow or make eggs?
- Mortality from starvation – don't all starve at once
- Spawning new superindividuals in bounded memory space

Methods

- “Fish” as modified floats
- Fixed number of fish per species per yearclass
- Limit number of yearclasses, killing off too old fish
- Feedback to NPZ-type model, NEMURO for now (PICES)
- Fish-eat-fish and fishing fleets require knowledge of fish in i,j space

ROMS Floats/Fish in MPI

- Each process has one tile, but knows about all floats
- Checks floats each timestep, zeroes out those not on tile
- Update floats on tile
- Sum reduction at end ensures all know about all floats

Eulerian vs. Lagrangian

- Lagrangian information is stored in a FISHES structure, which is a copy of the FLOATS structure with extra fields:
  - Logical :: alive, eatfish
  - Real :: bioenergy with weight, worth, age, birthday, yield, eggs, etc. components
  - Real :: feedback with NT components
  - Integer :: species, lifestage, swimtype

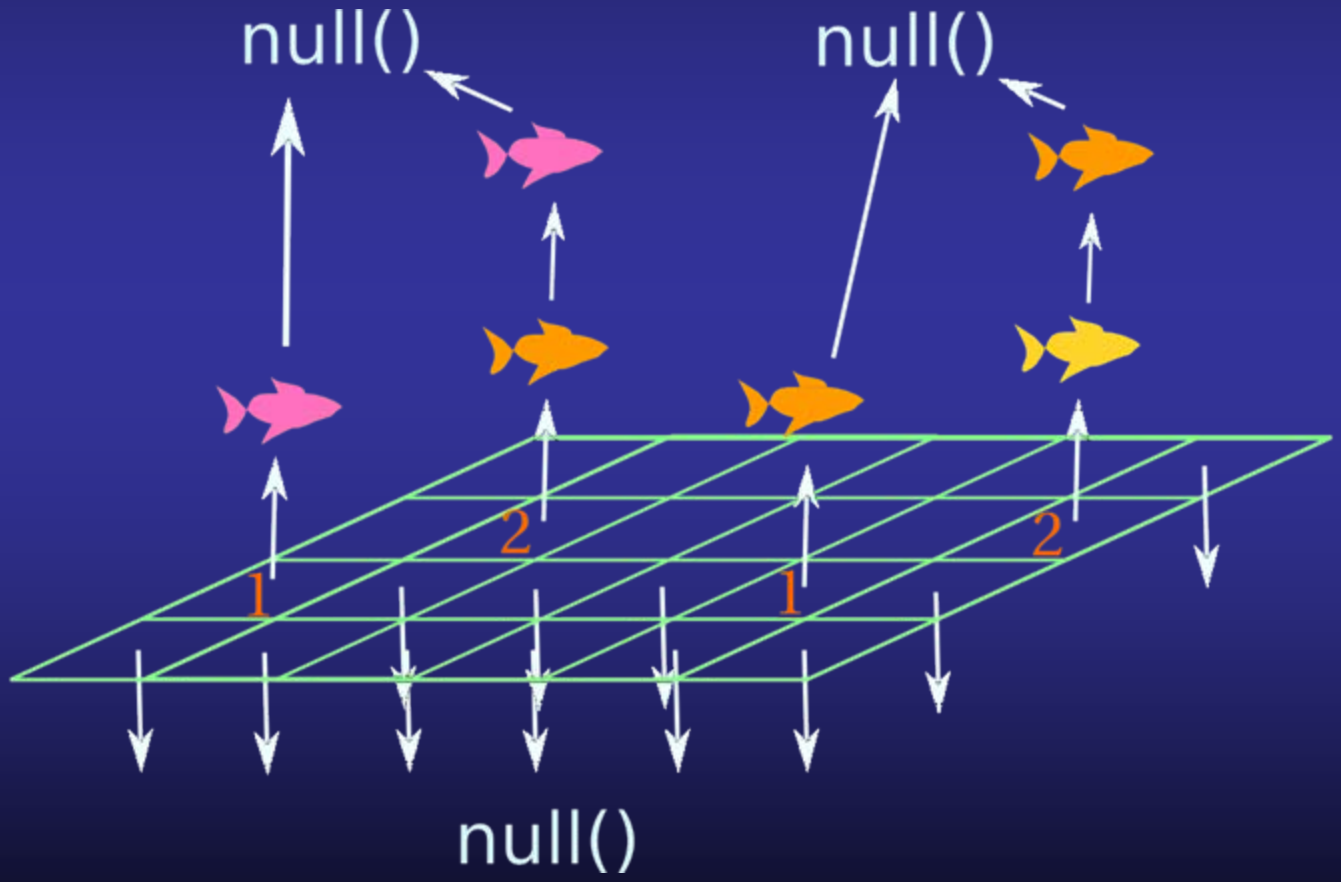
Eulerian Side

- Feedback to NPZ model happens in pfish array in nemuro.h:
  - Zero out pfish
  - Loop through fish (on tile), adding contributions to pfish
  - Loop through cells, modifying NEMURO fields according to pfish
  - Changes to zooplankton and PON

More Eulerian

- Mod_ocean.F contains fields on the tiles:
  - fish_count: number of fish in each 2-D cell
  - fish_list: pointer to first fishnode object in each cell (linked lists)
- For spawning (more later):
  - egg_count or spawn_dist

fish_list



Fish-eat-fish

- Loop through fish on tile – if I eat fish:
  - Look for other fish in cell
  - Are they the kind I eat?
  - There is cannibalism of eggs by adults
  - Look over whole water column for now
  - Worry about order? It's always going to be the same unless explicitly randomized

Behavior

- Eggs and yolk-sacs are advected like floats
- Larger fish swim with size-dependent speed
- Choice of behavior:
  - Humston: optimize temperature (kinesis)
  - Railsback: maximize growth (fitness)
- No spawning migration yet

Spawning

- Fixed number of new superindividuals available per year per species
- Set up beginning and ending dates on spawning season
- Divide up potential new fish over spawning days
- On each day, gather eggs and split into available new fish

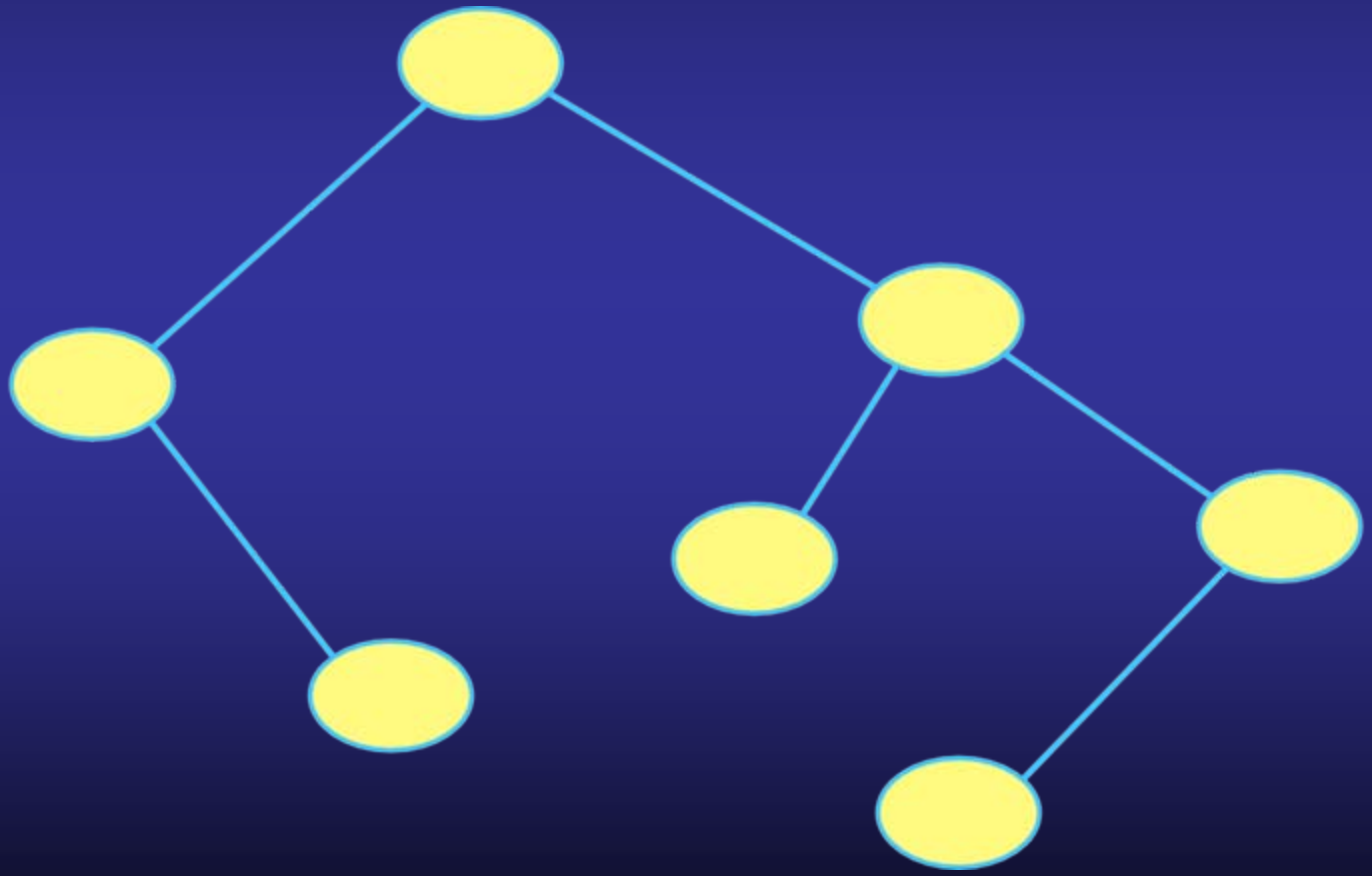
More Spawning

- If no spawning, put extras into pool
- Might have a few empties at year end
- Need at least one per spawning day – don't want to lose any eggs
- New superindividual retains position from one of the mothers' cells
- Choice of two batching methods...

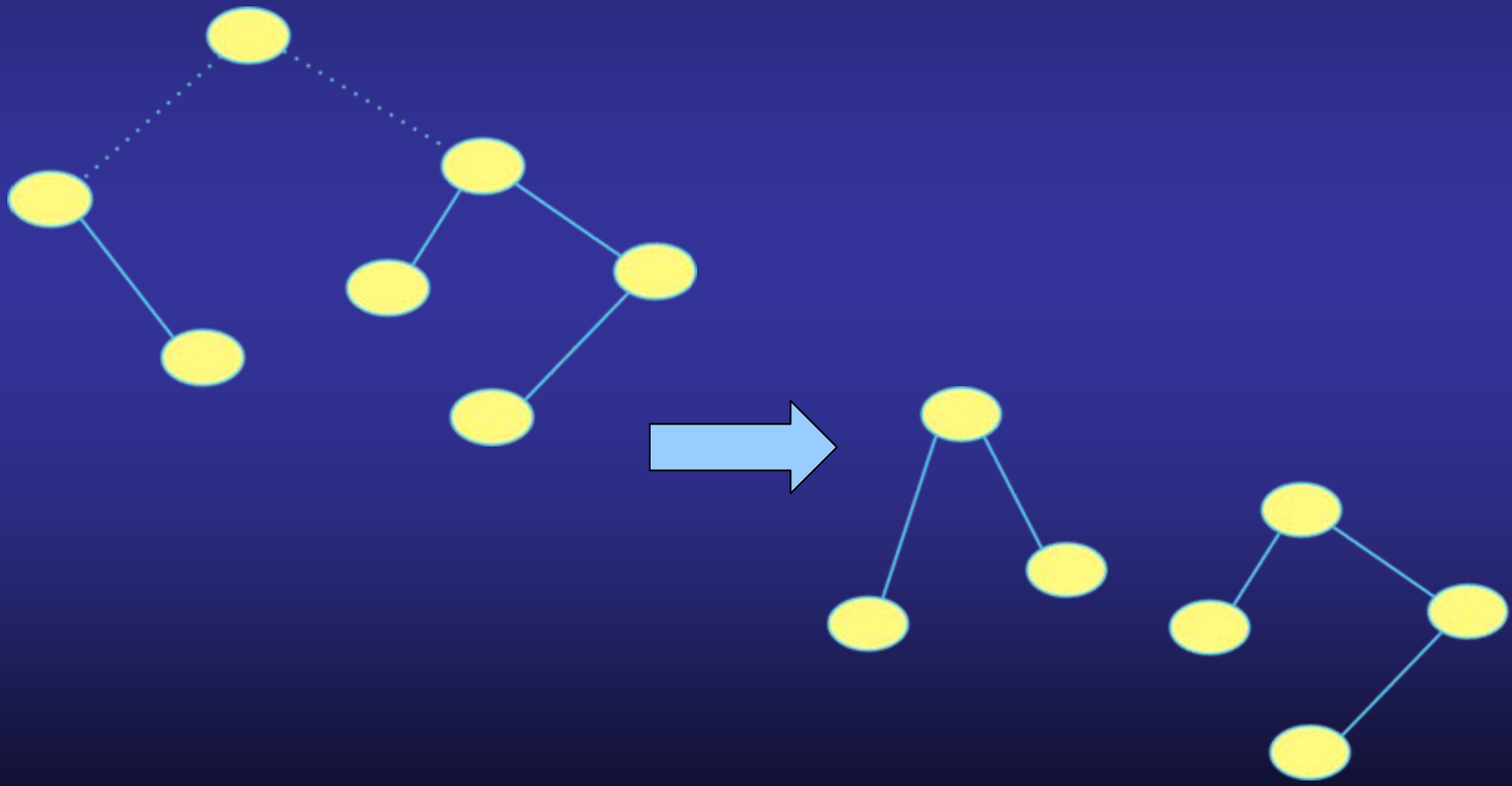
Binary Tree

- Master process builds a balanced binary tree from the spawning parents, in sorted order
- One tree node per spawning fish
- Method depends on a 1-D sorting order, such as long-shore distance
- Spawn_dist field comes from grid file or analytic function of I,j

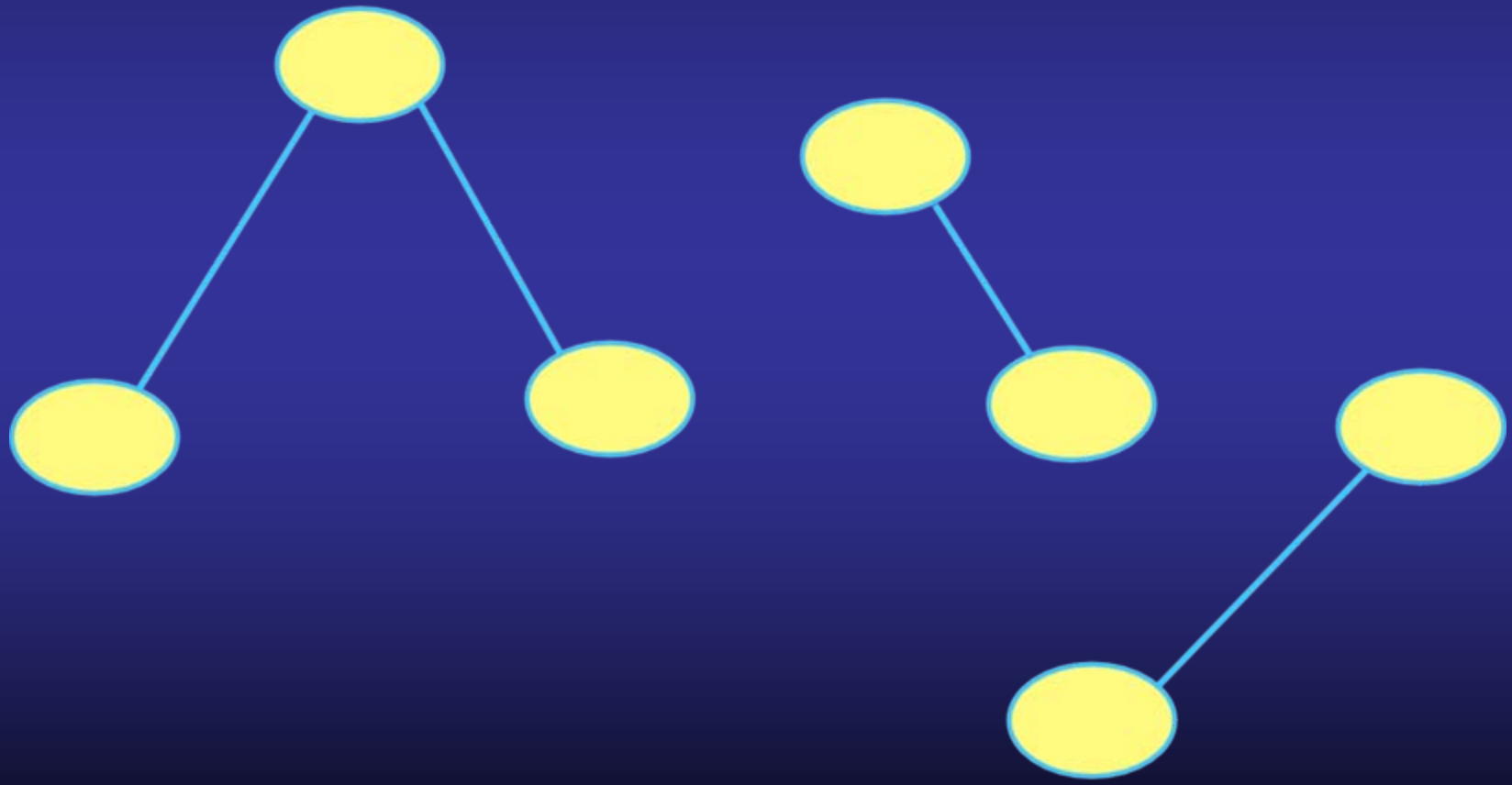
Balanced Binary Tree



First Split: Top node added to left subtree which rebalances



Final Three Subtrees



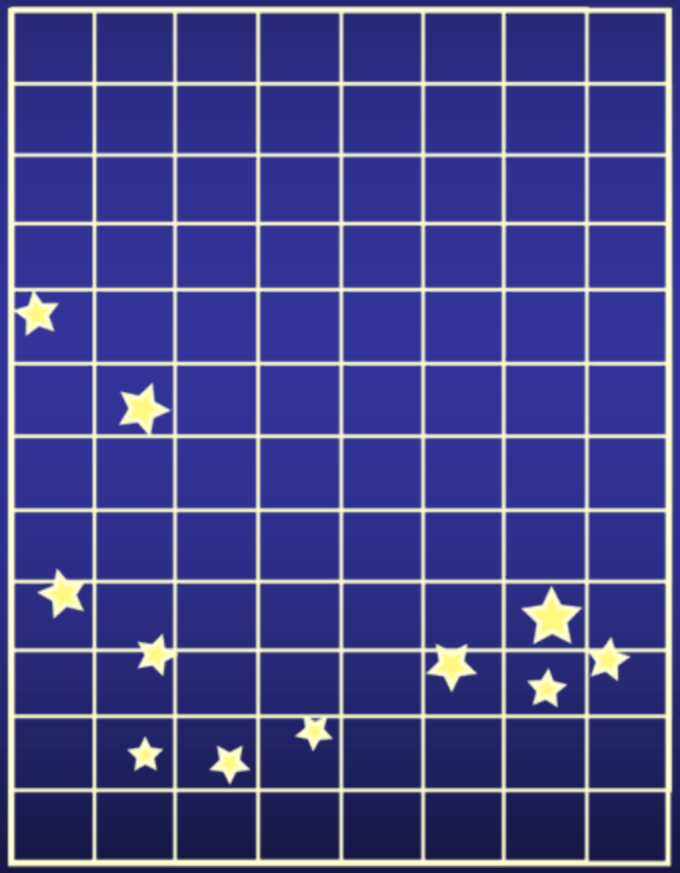
Create New Superindividuals

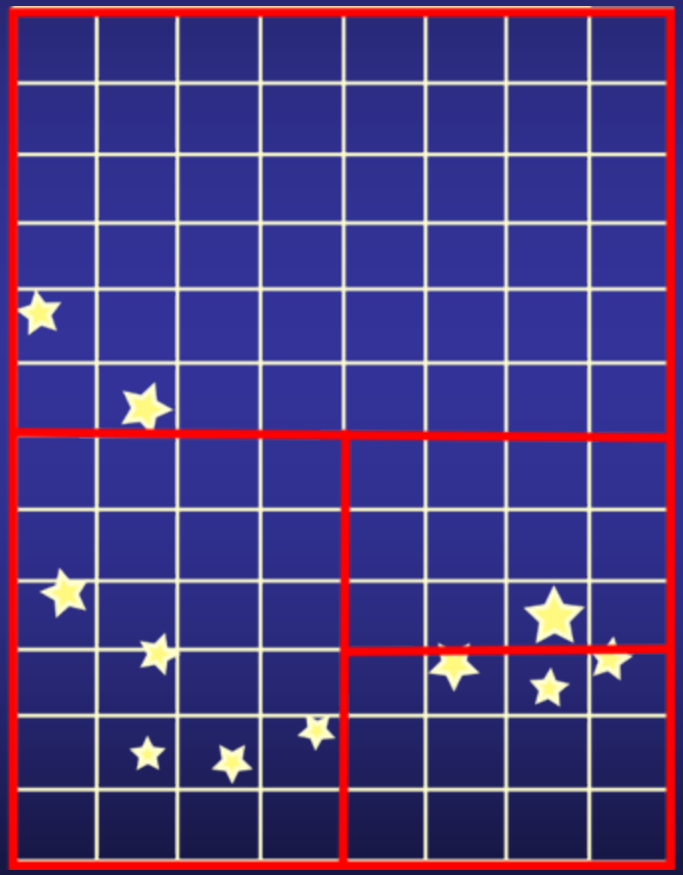
- Take the top of each subtree, find location of the parent fish
- Assign location to new fish
- Assign all the eggs to new fish
- New fish has lifestage of egg
- All happens with the FISHES structure, no knowledge of grid

Bisection

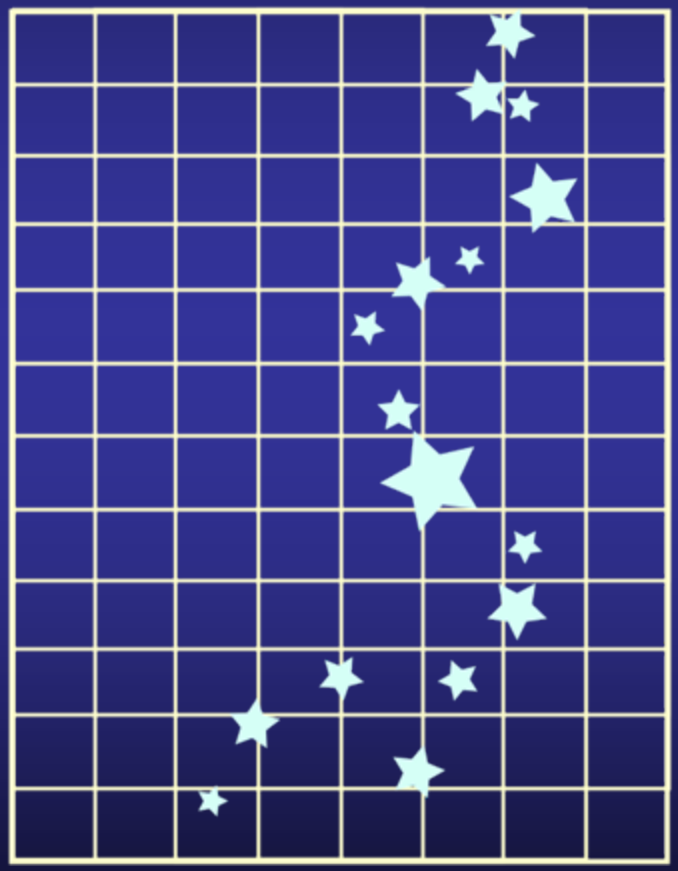
- Master node builds an egg array with all of the eggs in each cell
- Array is for the entire grid
- Successively divide up domain in I,j directions until available superindividuals are filled
- Toss out empty partitions
- Keep a sorted linked list of partitions

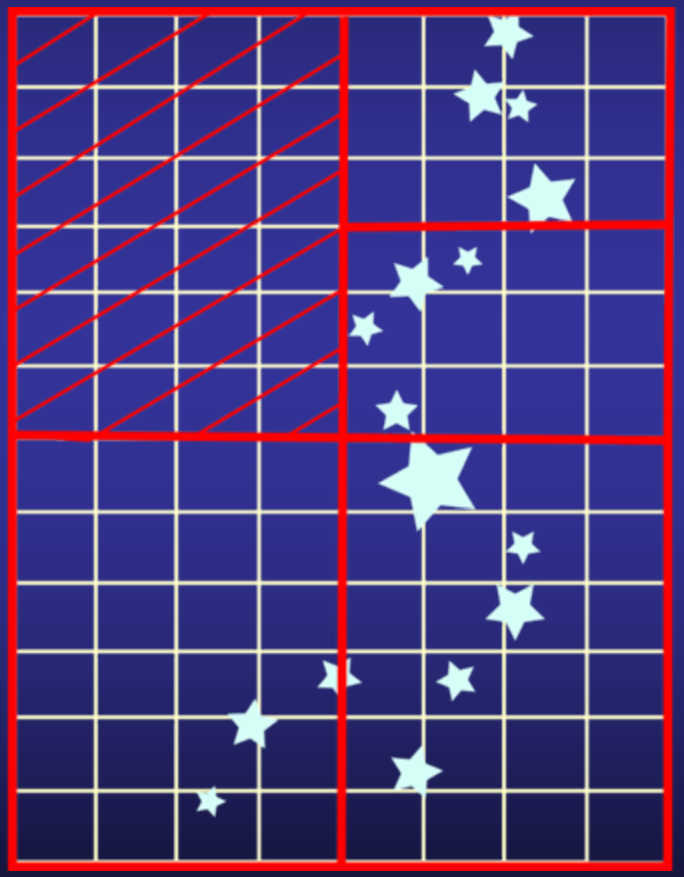
Bisection

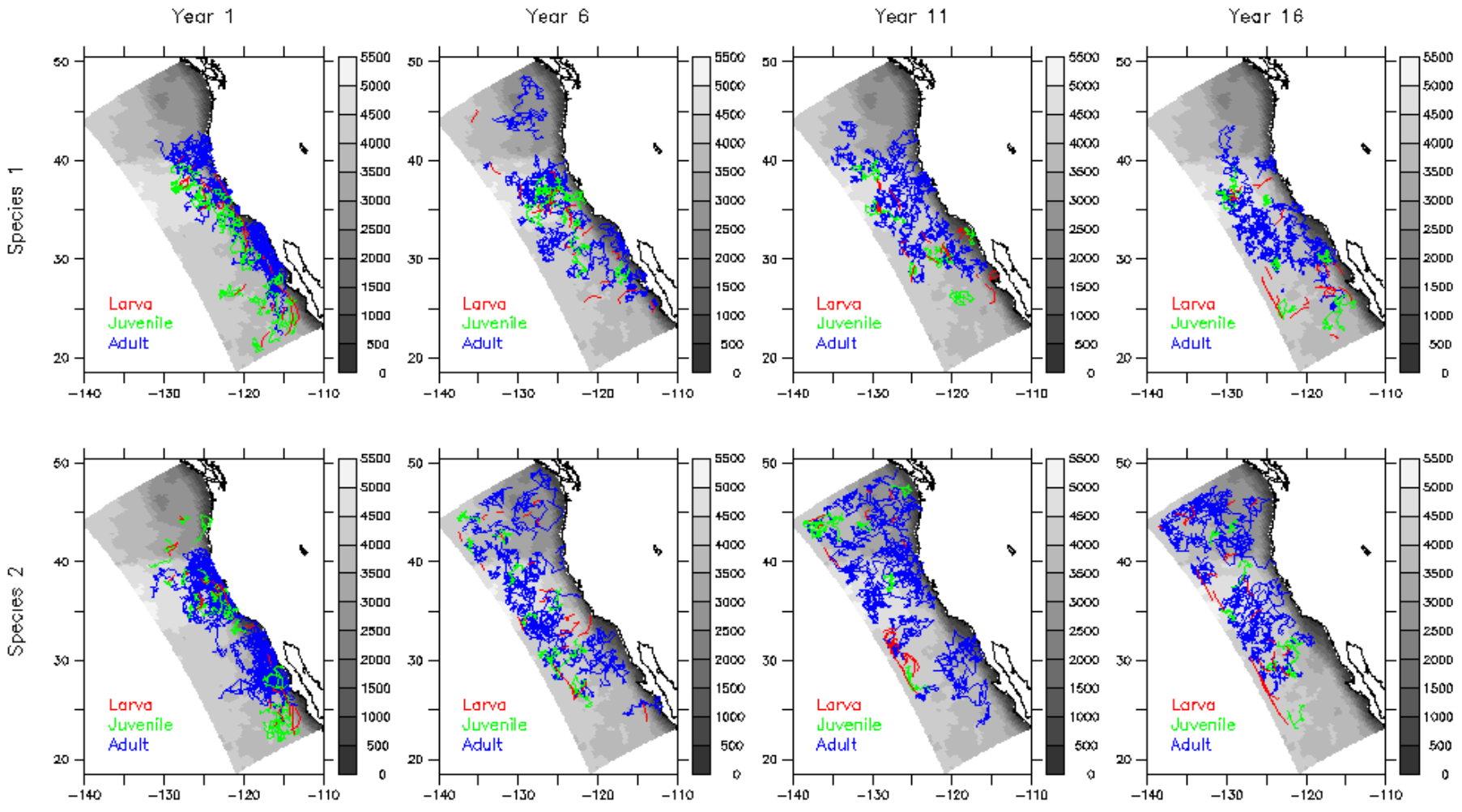


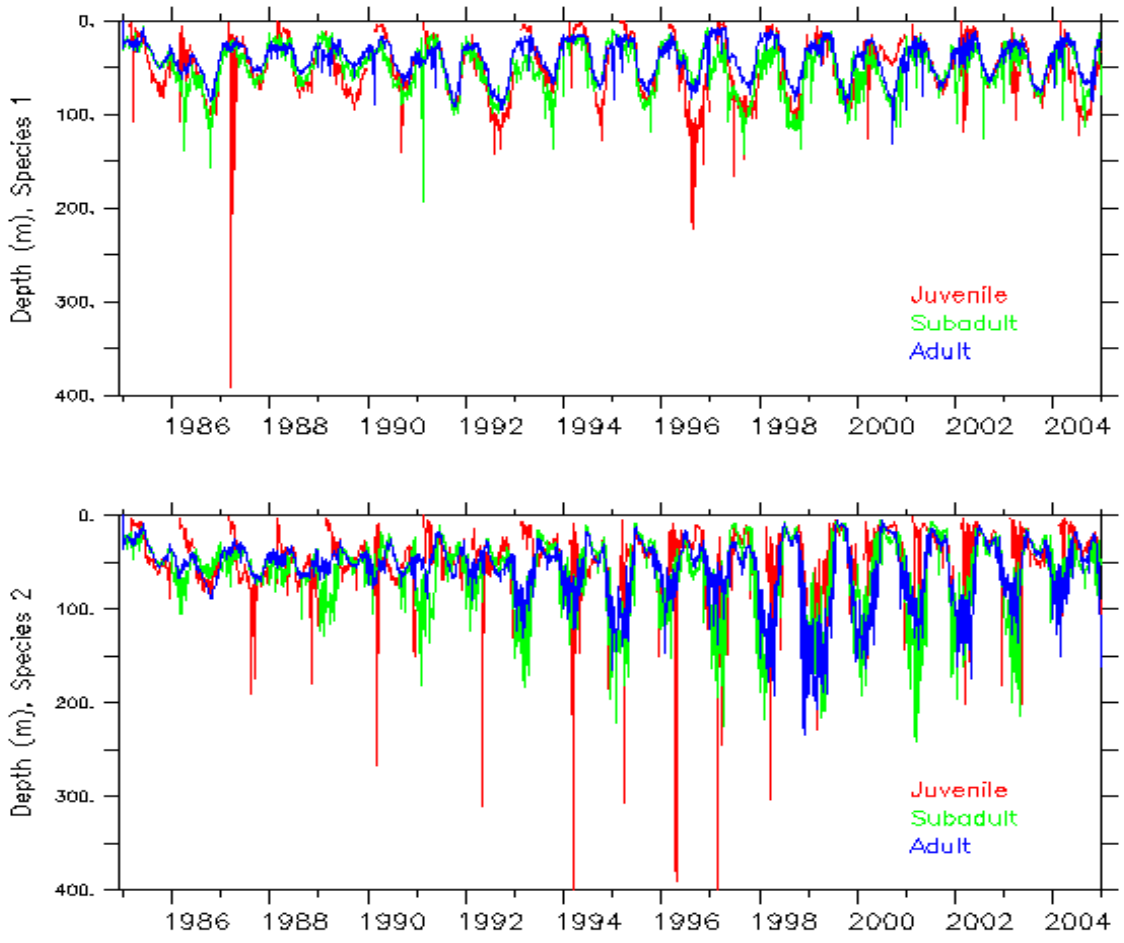


Another Example






Year 1
Year 6
Year 11
Year 16
Species 1
Species 2
Larva
Juvenile
Adult
50
40
30
20
−140
−130
−120
−110
5500
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0


Depth (m), Species 1
0
100
200
300
400
1986
1988
1990
1992
1994
1996
1998
2000
2002
2004
Juvenile
Subadult
Adult
Depth (m), Species 2
0
100
200
300
400
1986
1988
1990
1992
1994
1996
1998
2000
2002
2004
Juvenile
Subadult
Adult

Conclusions

- We are well on the way towards having a full lifecycle model of fish in ROMS
- These fish represent small pelagics, up to 5-10 species
- Depends on bioenergetics data
- Need to “prime the pump” with adult fish, then run for decades

Future Plans

- OFFLINE_FISH option [soon]
- Fishing fleets and predators
  - Behaviors
  - Fixed numbers, no spawning
- Tuning of fish bioenergetics and behavior
  - Warm/cold behavior switch
- Look to neighboring cells