Automated grid generation for WAVEWATCH III®

Arun Chawla
SAIC @ Marine Modeling and Analysis Branch
NOAA / NWS / NCEP / EMC

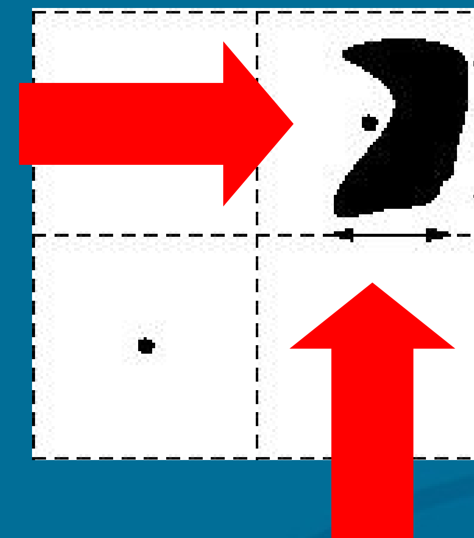
Arun.Chawla@NOAA.gov

Motivation

- WAVEWATCH III® requires 3 grids (1 necessary and two optional)
  - A bathymetry grid (necessary)
  - A land –sea mask grid (optional in WW III v 2.22, needed in multi-grid version of WW III)
  - Obstruction grid to account for energy decay due to sub-grid blocking (optional)
- Development of these grids can be a fairly arduous task. This is specially true for multi-grid version of WW III where grid consistency across overlapping grids is necessary for accurate two – way coupling.
- To facilitate this we developed a set of algorithms that can automatically create accurate grids with minimal input from the user.
- The necessary tools have been developed using MATLAB.

Obstruction grid – Proof of concept

Tolman (2003) showed that sub-grid islands can be modeled in WAVEWATCHIII by physically reducing the energy fluxes between the cells



<u>1D Spatial propagation in WAVEWATCHIII</u>

$$F_i^{n+1} = F_i^n + \frac{\Delta t}{\Delta x}\left(\alpha_{i,-}G_{i,-} - \alpha_{i,+}G_{i,+}\right)$$

Spectral density

Density flux and transparencies at cell boundaries

Reduction of energy dependent upon the proportion of cell being obstructed

Obstruction grid ranges from 0 (no obstruction) to 1 (full obstruction)

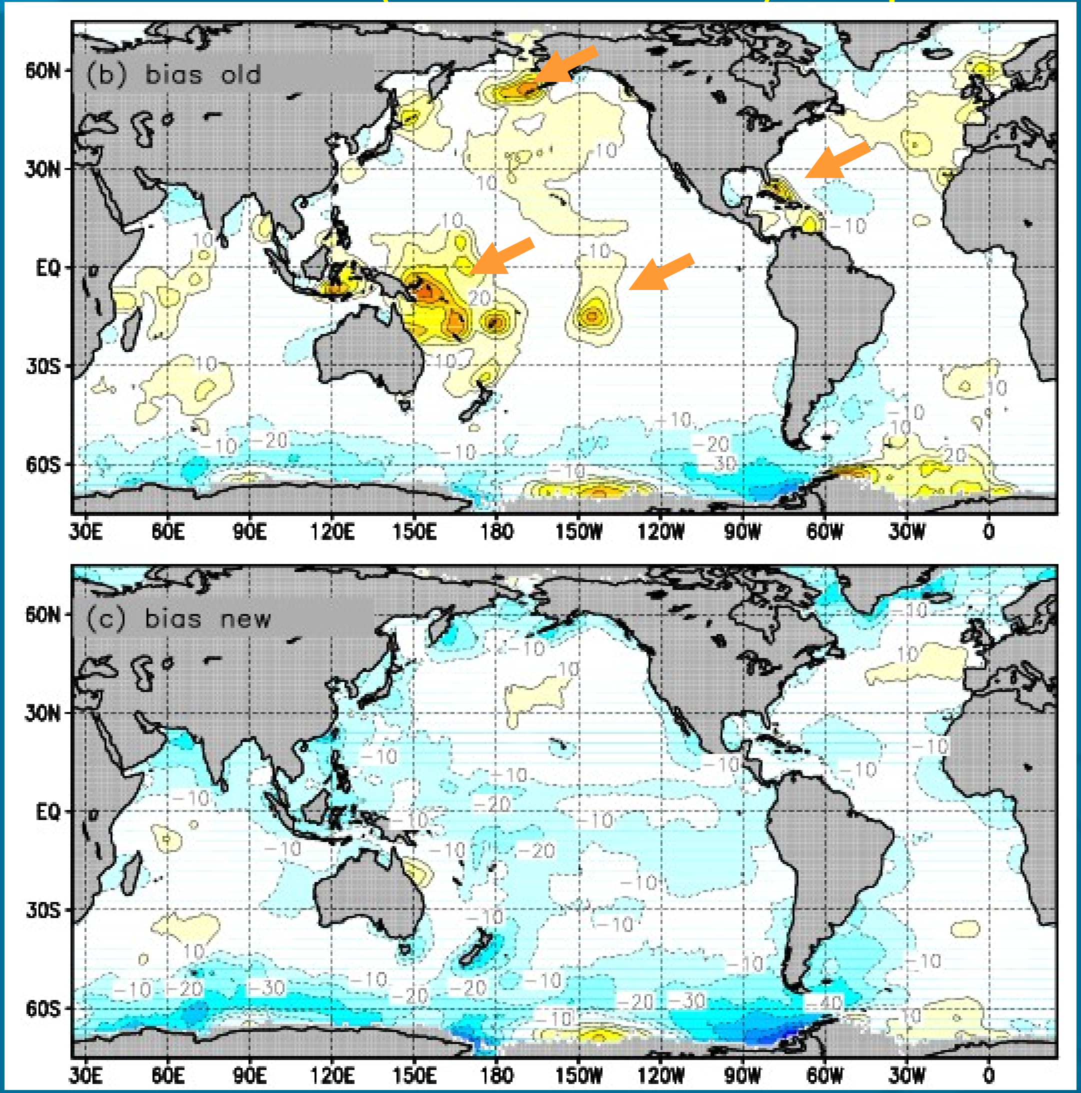
Two obstruction grids (for the 2 directions of motion) used in WAVEWATCHIII

Obstruction grids in action

Bias (Model –Data) map

No obstruction grid

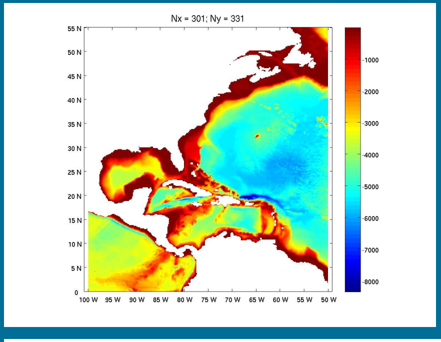
Obstruction grid



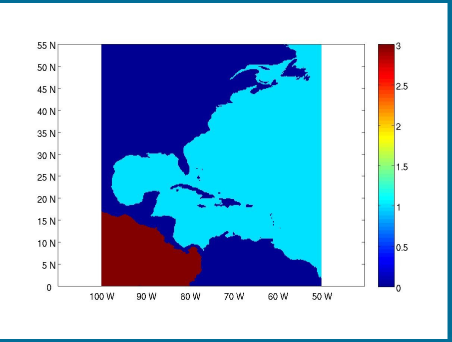
Obstruction grids remove the bulls – eye patterns behind islands

MODULES

Step 1: Build a bathymetric grid from a high resolution base bathymetry

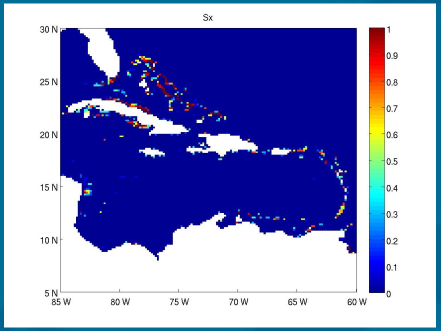


Step 2: Create an appropriate land / sea mask to accurately depict coast lines



Step 3: Mask out un-necessary water bodies

Step 4: Generate obstruction grids for unresolved islands



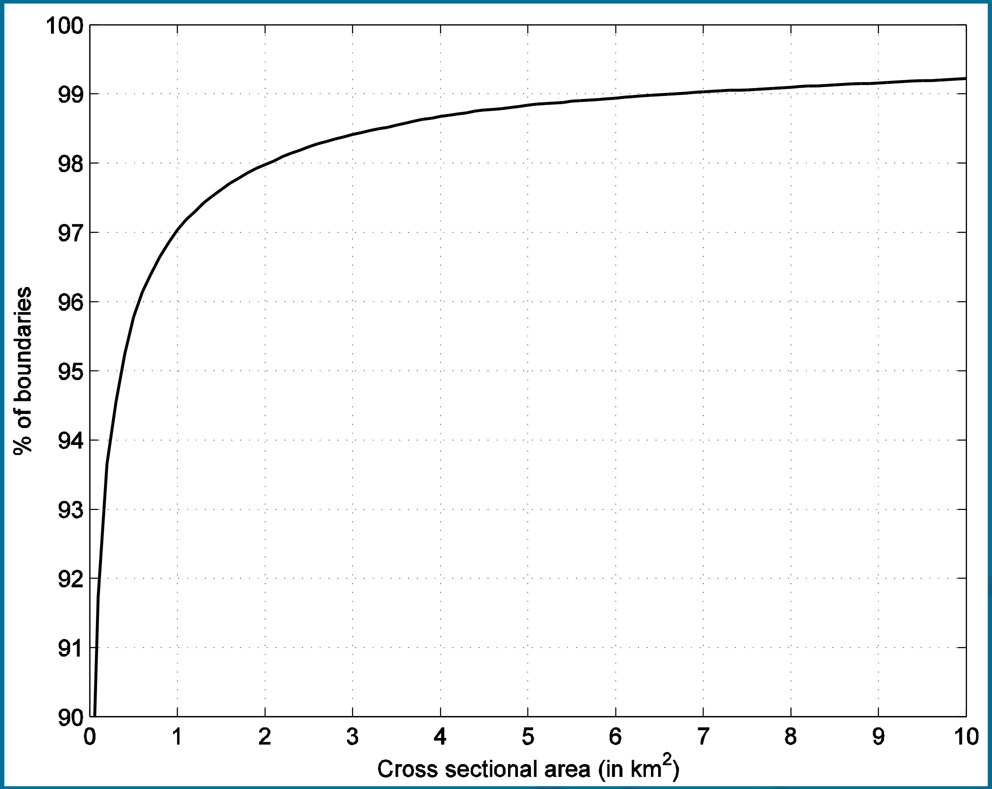
Step 5: Identify nodes for exchanging boundary information (multiple-grid only)

Reference Data

- Two types of high resolution reference data available
  - A global high resolution bathymetry data set
    - ETOPO2 from the National Geophysical Data Center
      - 2’ arc length global relief bathymetry data set
    - ETOPO1 from the National Geophysical Data Center
      - 1’ arc length global relief bathymetry data set
  - A global shoreline database in the form of polygons (**GSHHS** - **G**lobal **S**elf-consistent **H**ierarchical **H**igh-resolution **S**horeline)
- Algorithms will be designed to meld the high resolution bathymetry with the shoreline database to develop the optimum grids.

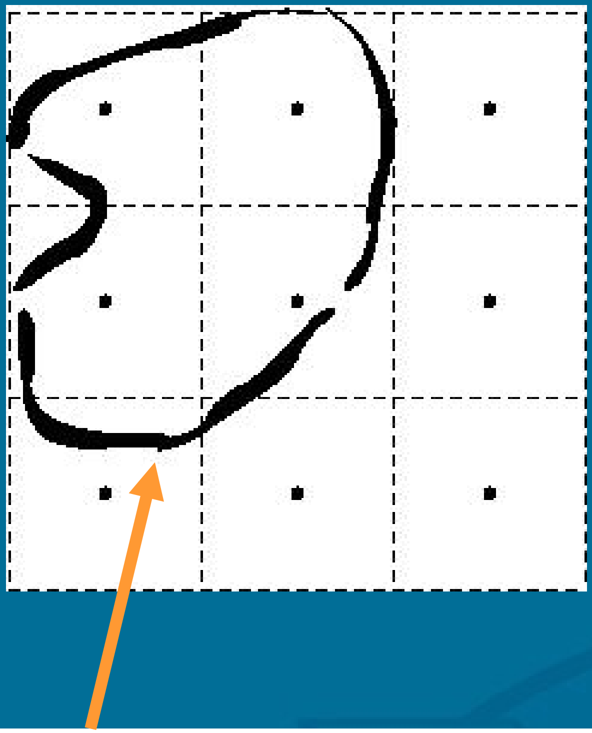
Why Shoreline Polygons?

- There are 188,606 shoreline polygons (180,509 coastal) in the data base
- Over 99 % of these have a cross sectional area < 6 km2 (cross sectional area of a 2' grid square ~ 14 km2)
- Convenient to treat land bodies as closed polygons
  - Precludes need for representation in high resolution grid
  - Trivial to compute extent of coastal bodies along the grid axes



Why Shoreline Polygons (contd.)?

- Atolls are very well represented
- Additional obstructions (e.g. breakwaters) easily added
- Trivial to mask out selected bodies of water (e.g. Hudson Bay) or reefs (e.g. Great Barrier Reef)



Atolls cover very little surface area but provide effective barriers to wave propagation

Subroutines

- **A grid generation routine**

  Generates a grid from base bathymetric data set. We use global ETOPO2/ETOPO1 bathymetry netcdf files. (NOTE: To use this routine your matlab should be able to handle netcdf files. Capability is available as open source)

- **A boundary extraction routine**

  Extracts a subset of boundaries from a global set of polygon boundaries (GSHHS)

- **A land-mask routine**

  Blends the bathymetric data with the coastal boundaries to develop an accurate land-sea mask

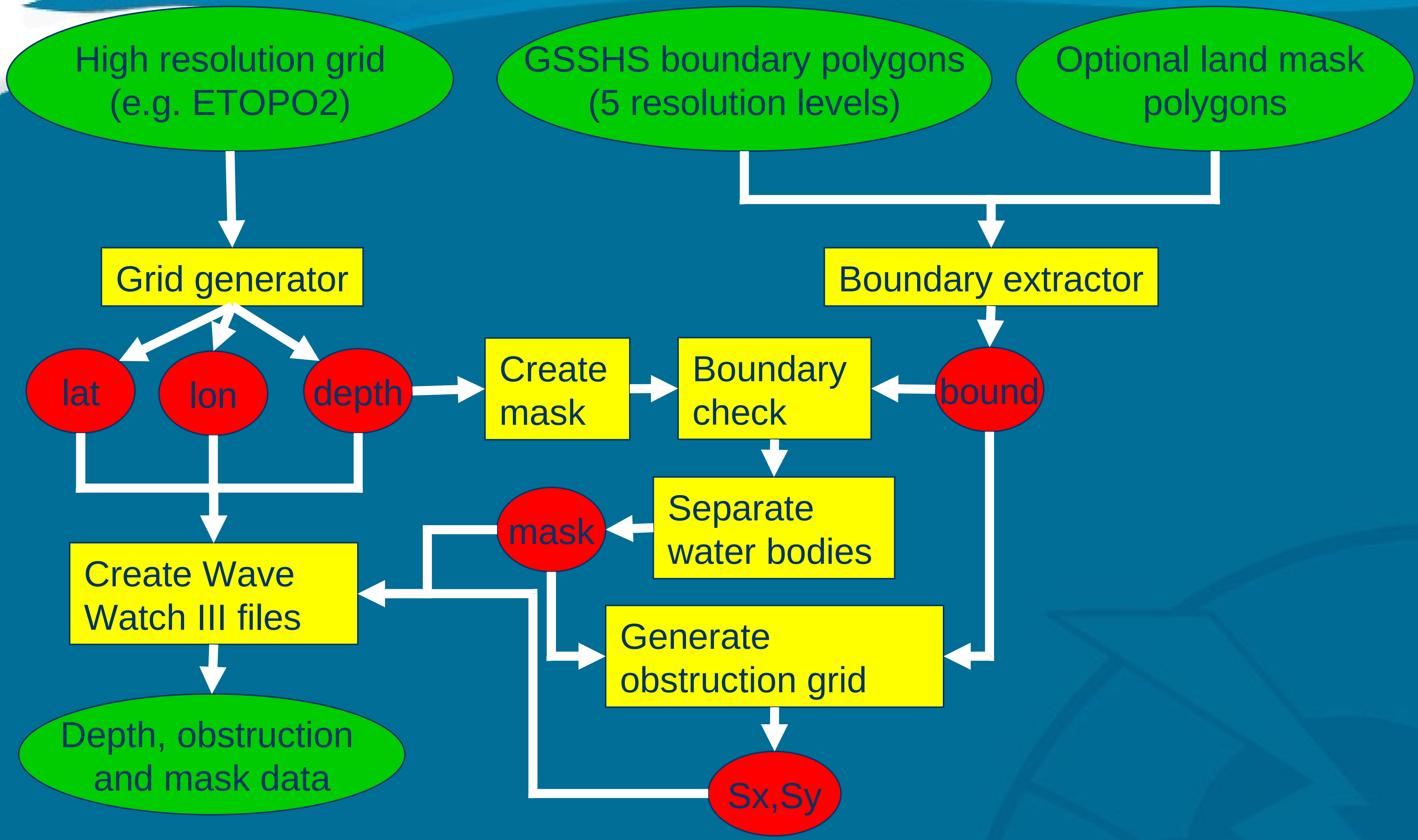
- **A water body routine**

  Groups the wet cells into different water bodies (each having a unique id)

- **A sub-grid obstruction routine**

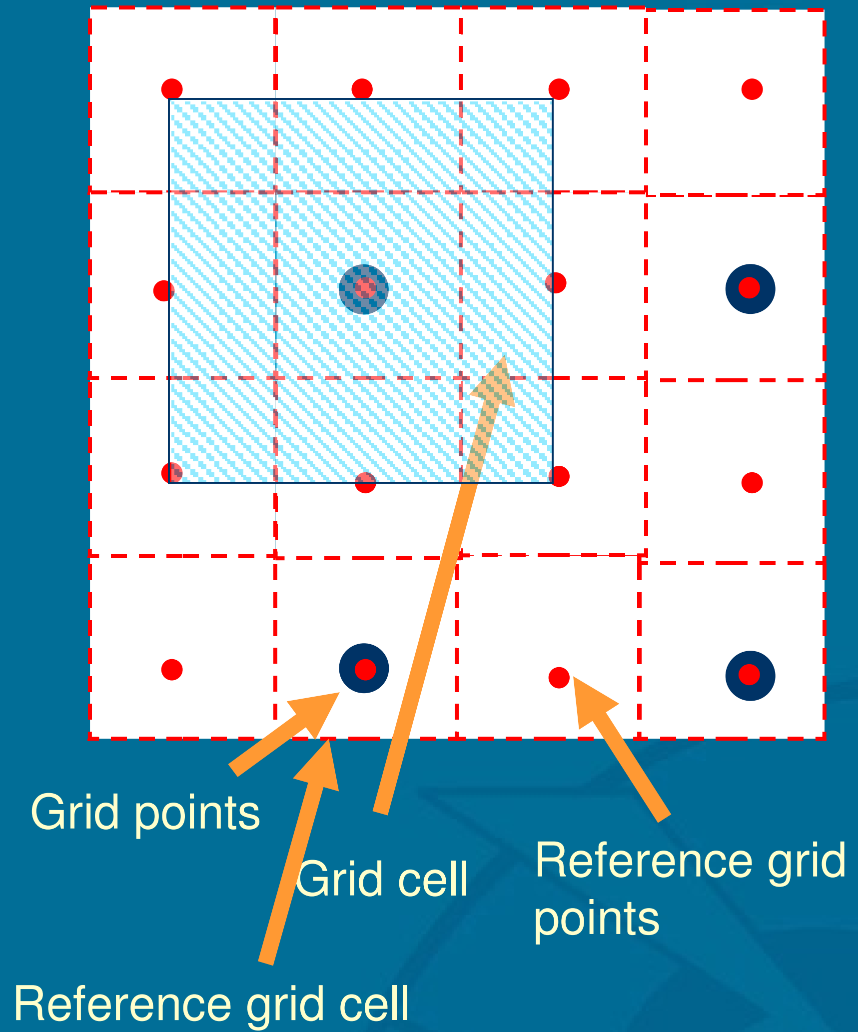
  Develops sub-grid obstruction matrices in x and y direction for wet cells

Flow Chart


High resolution grid (e.g. ETOPO2)
GSSHS boundary polygons (5 resolution levels)
Optional land mask polygons
Grid generator
Boundary extractor
lat
lon
depth
Create mask
Boundary check
bound
Separate water bodies
mask
Create Wave Watch III files
Generate obstruction grid
Depth, obstruction and mask data
Sx,Sy

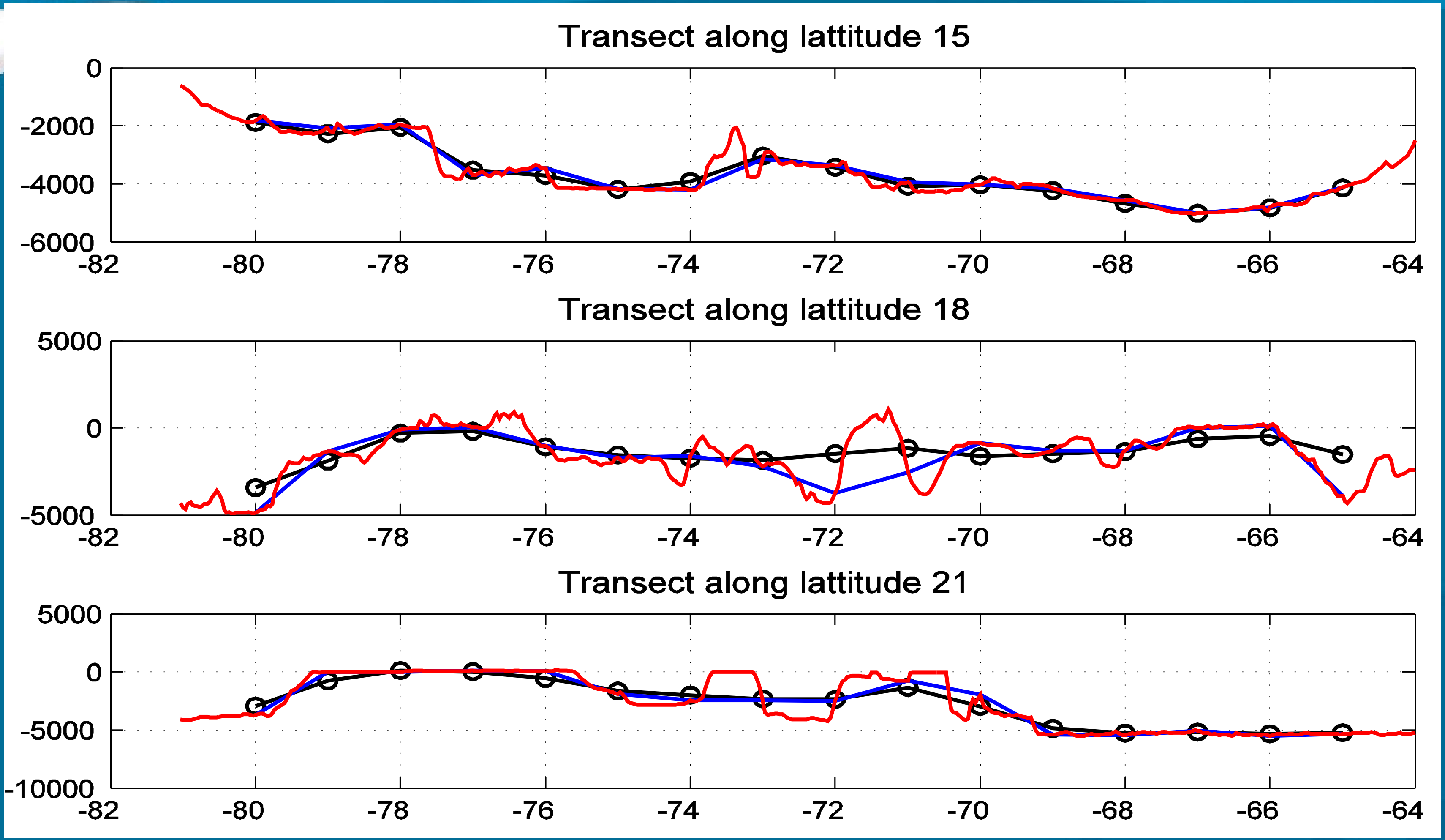
Grid Generation Routine

- Routine uses 2D averaging to interpolate the higher resolution (reference) grid at the lower resolution
- Averaging carried out over all the reference “wet cells” that lie within a grid cell
- If the proportion of reference wet cells is less than user specified cut-off (ranging between 0 and 1) then the grid cell is marked dry.
- Averaging filters out higher spatial frequencies and prevents aliasing



Grid Generation (contd.)

Bathymetry cross sections along 3 transects in the Bahamas



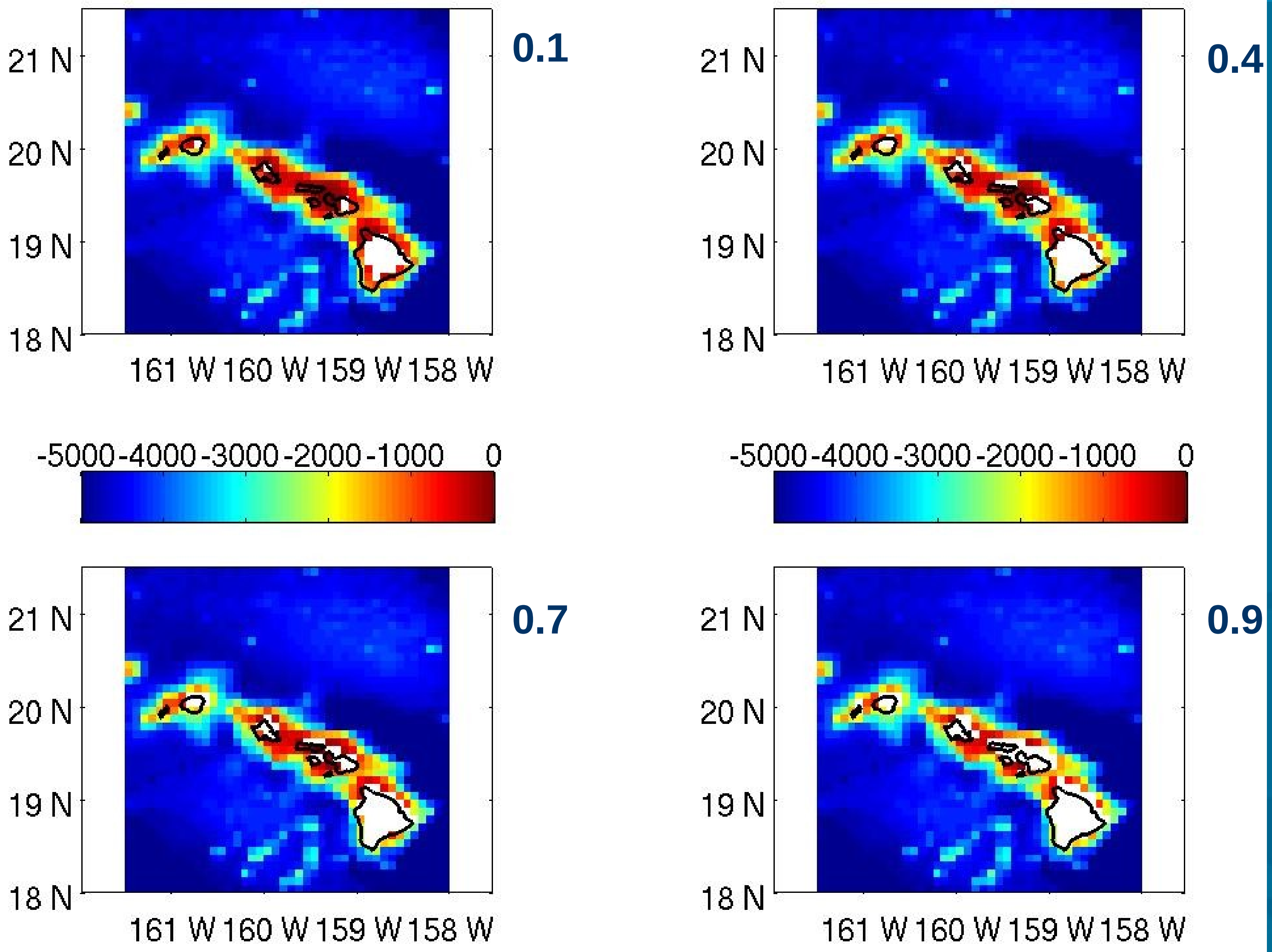
Averaged bathymetry

Sampling points

High resolution bathymetry

Sub sampled bathymetry

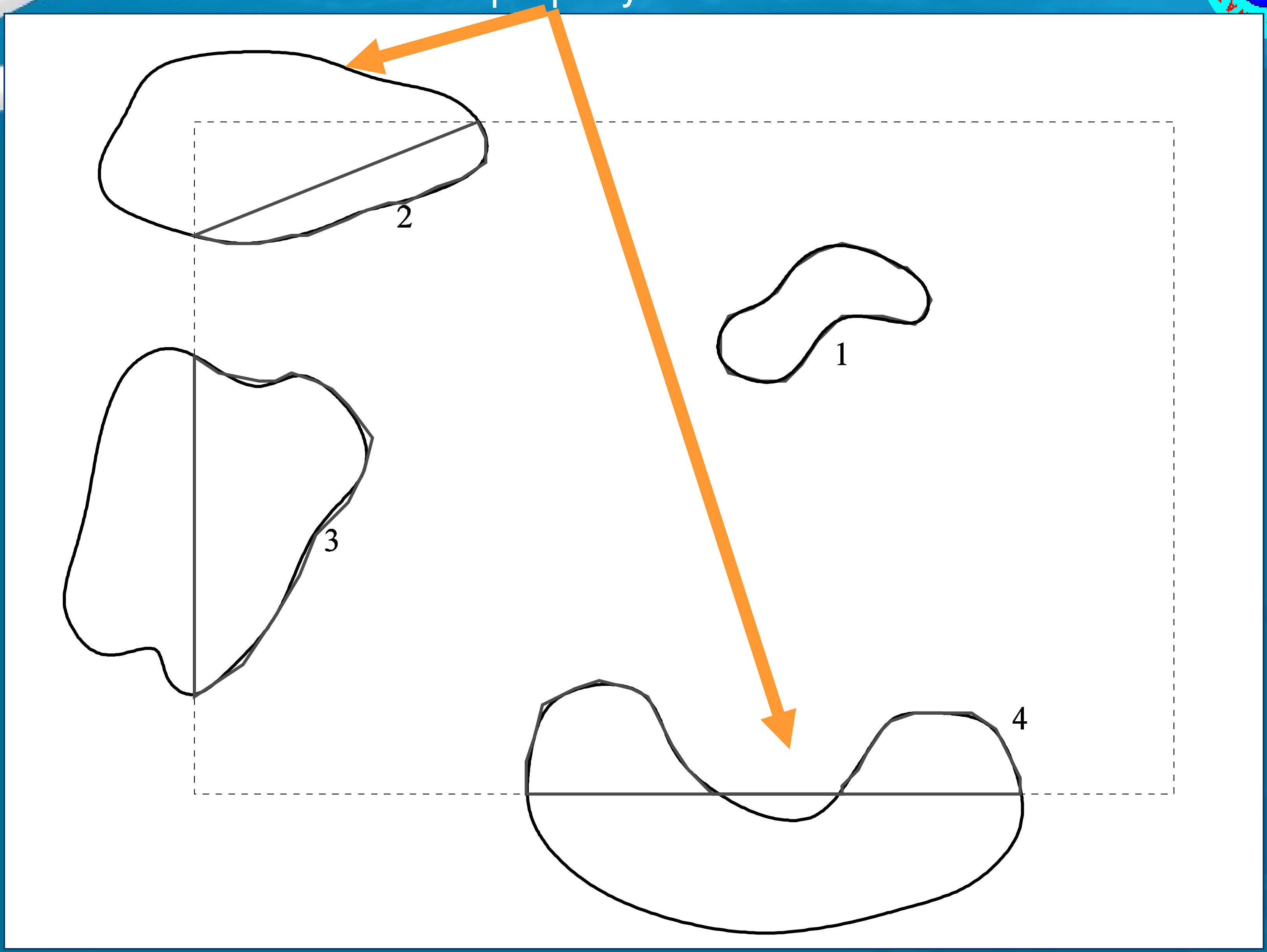
Bathymetry (Hawaiian islands) using different cutoffs for proportion of wet cells



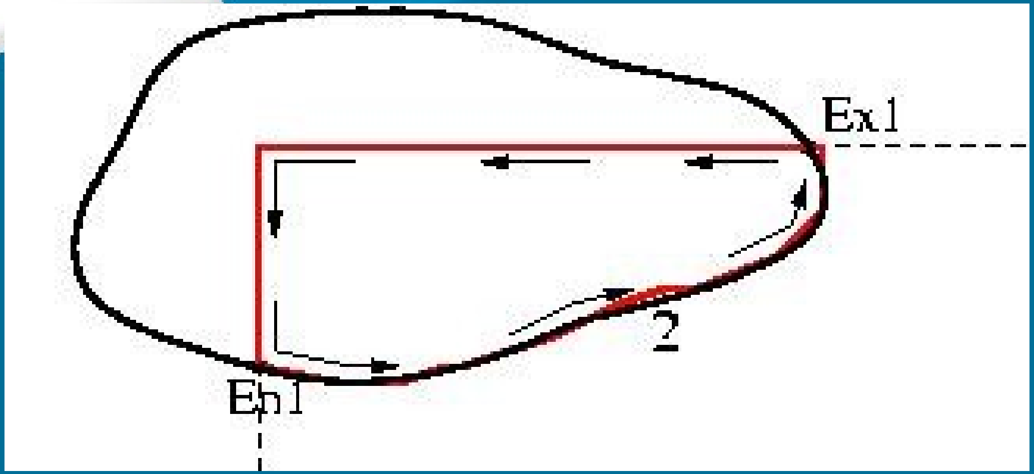
Boundary extraction routine

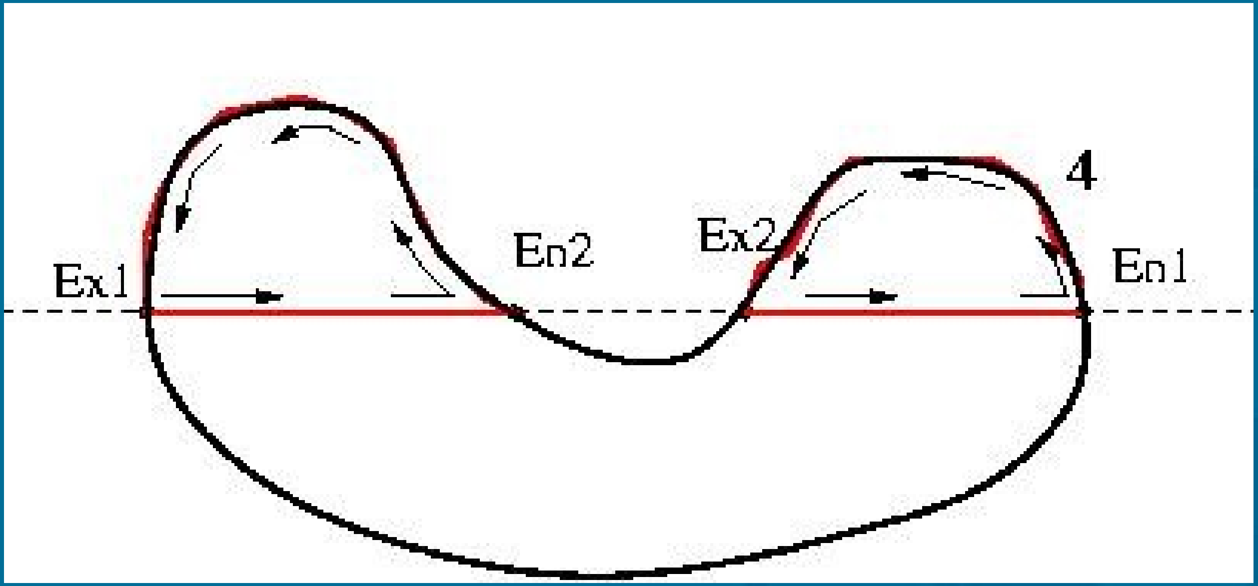
- Uses the GSHHS (Global Self-consistent Hierarchical High-resolution Shoreline) polygon database.
- High resolution data is available as mat files at 5 resolutions
  - Full
  - high (0.2 km)
  - intermediate (1 km)
  - low (5 km)
  - coarse (25 km)
- Only accounts for land – sea boundaries (ignores lakes)
- Properly splits boundaries that are intersecting grid domain boundary
  - Important to properly close boundaries for land masks and sub grid obstructions

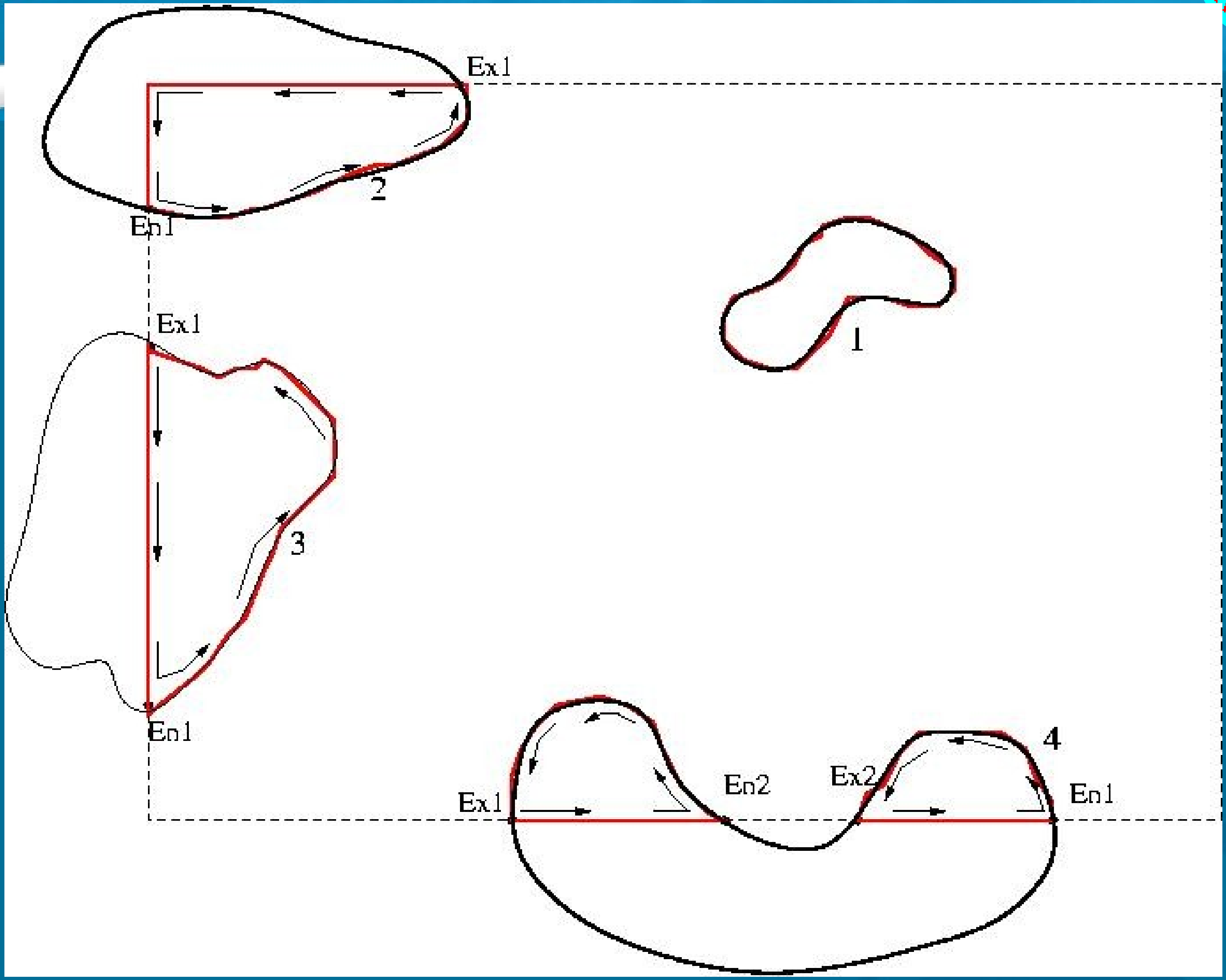
Improperly closed boundaries



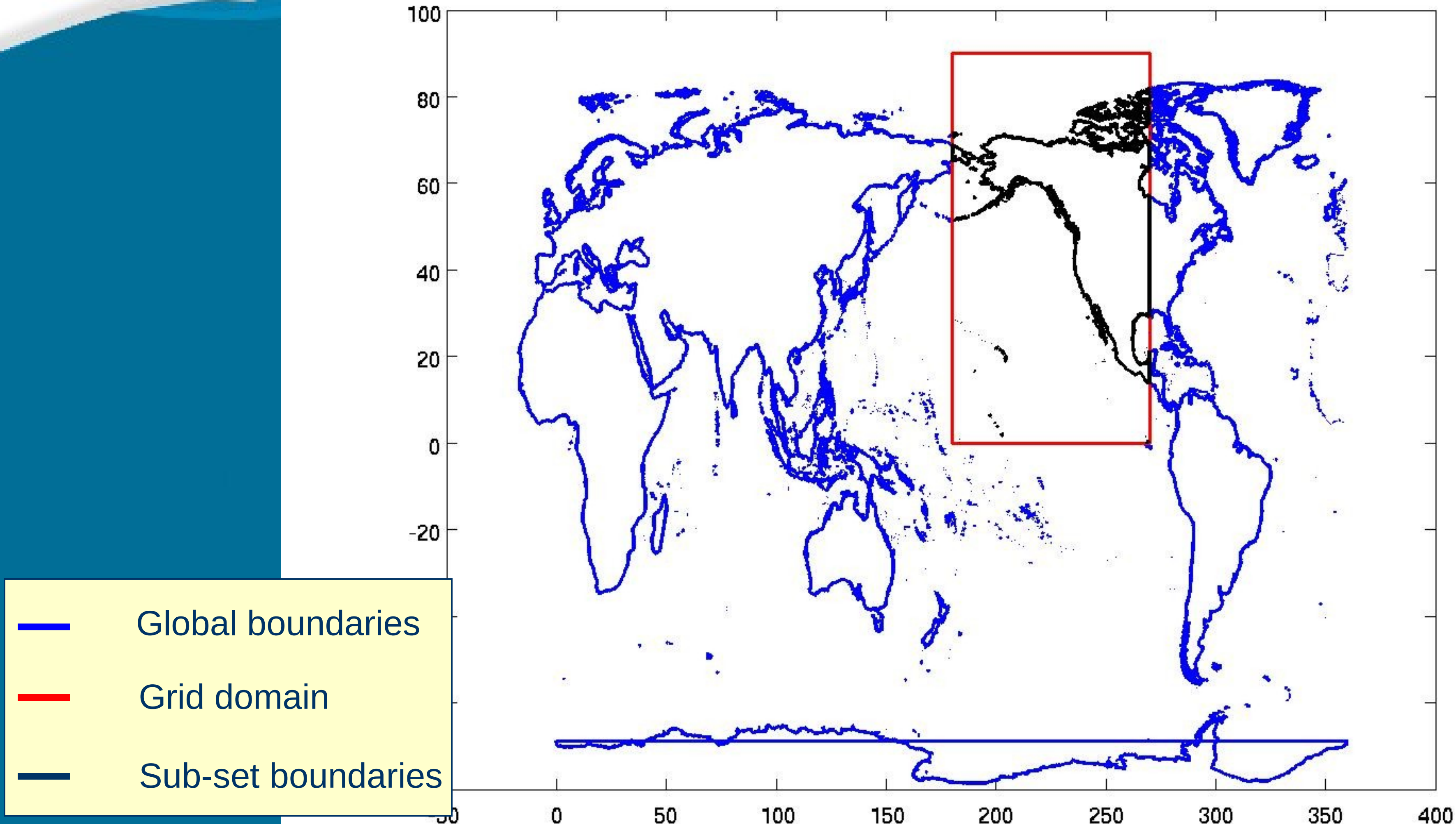
Splitting a boundary properly






Ex1
2
En1
1
Ex1
3
En1
4
Ex1
En2
Ex2
En1

Full Resolution Global Boundary

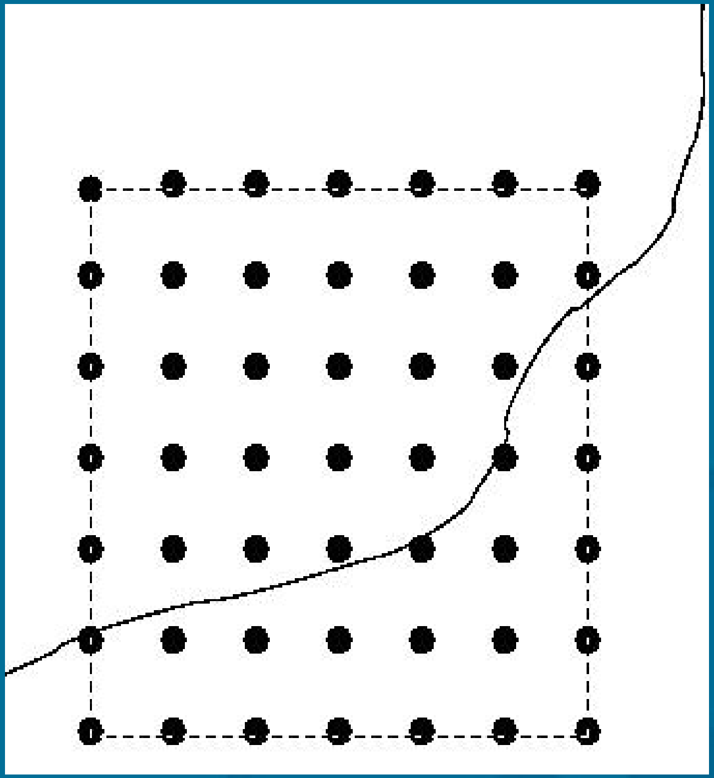


Land-sea mask routine

- A first order land mask (based on a depth cutoff) is determined
- The land mask routine then
  - Checks all wet cells near the boundaries
  - Switches wet cells to dry if a proportion of cell within the boundary > user specified cutoff (currently set at 0.5)
- Land mask routine needed to
  - Account for land masses not present in base bathymetry grid
  - Resolve discrepancies between shoreline data base and base bathymetry shorelines
  - Account for additional polygons

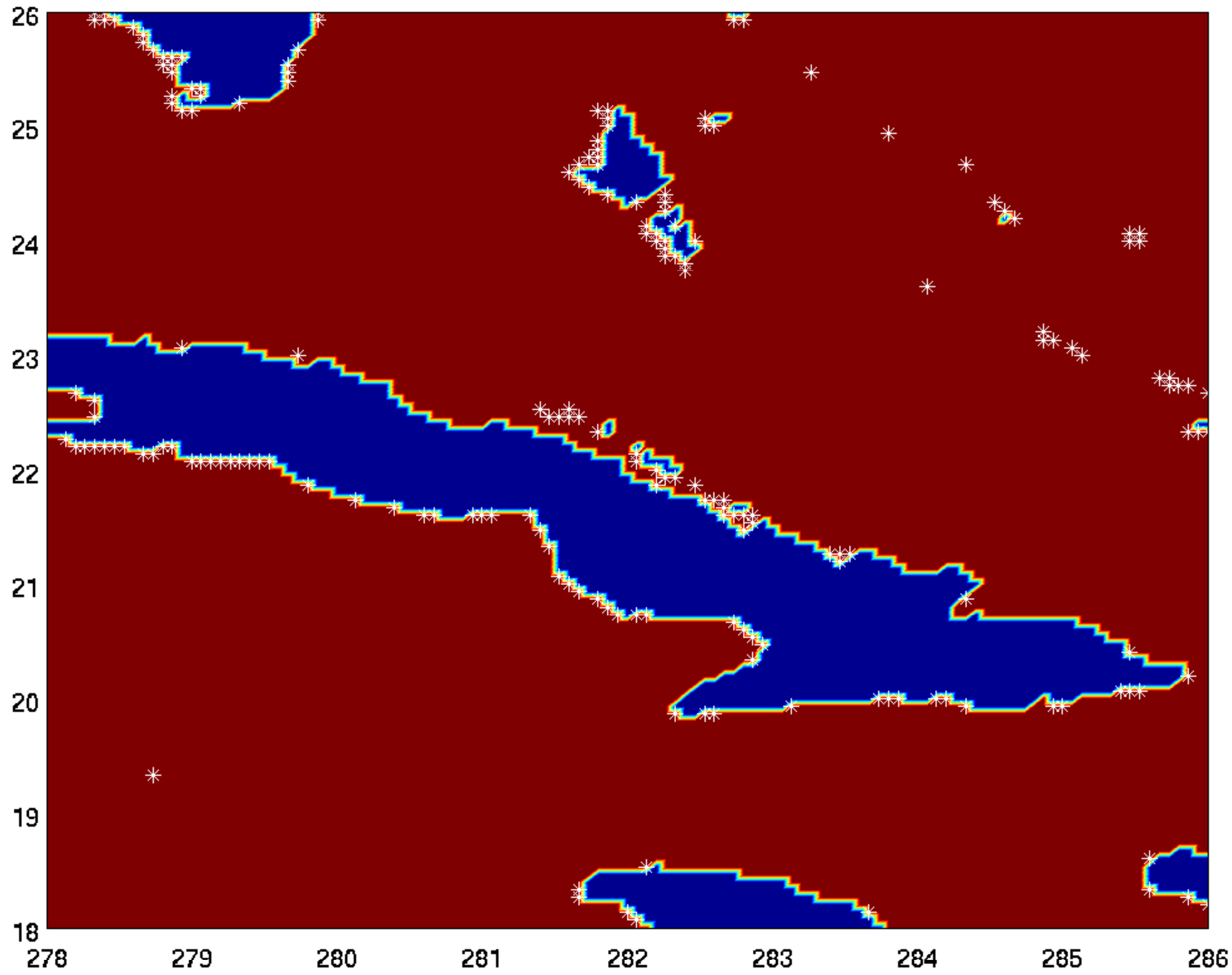
To compute proportion of cell inside boundary

- Equally spaced points spread throughout the cell
- Determine number of points enclosed within boundary
- Number of points inside boundary/Total number of points

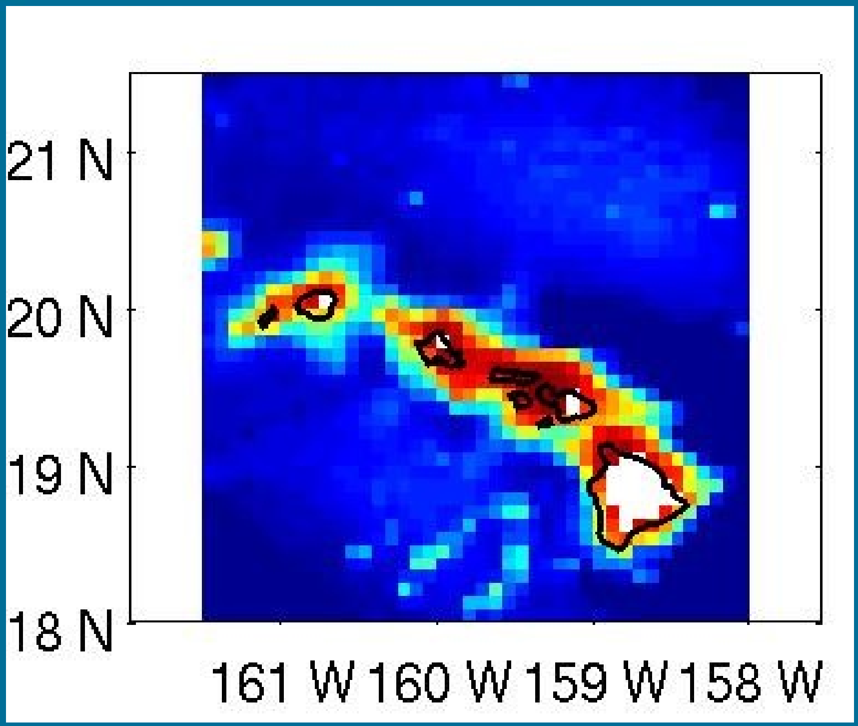


Land-Sea Mask (Bahamas 15 min grid)

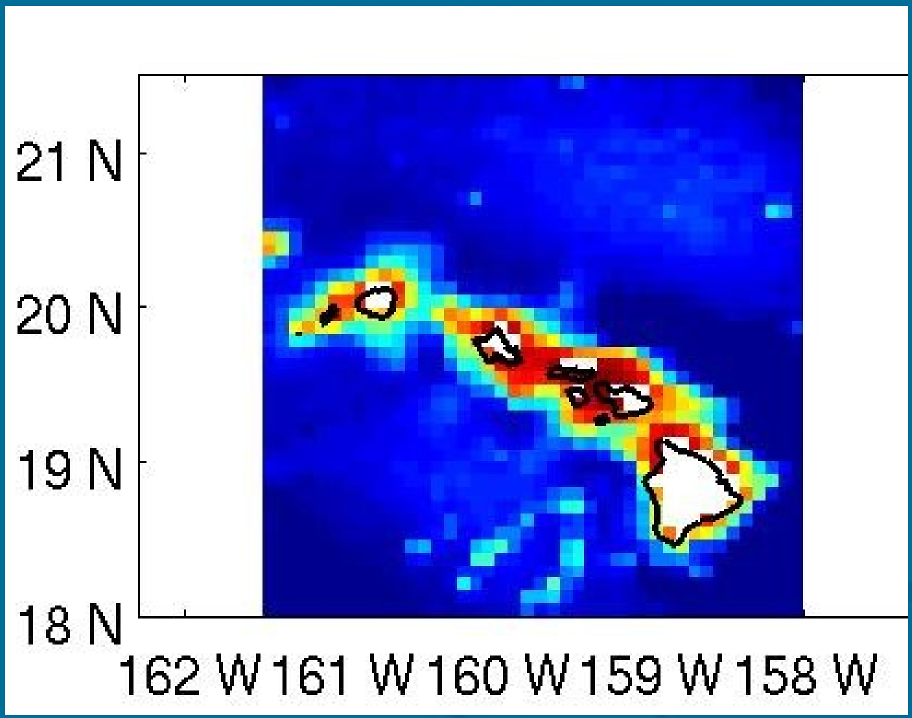
Red – Wet Cell
Blue – Dry Cell
White – Wet cell switched to dry cell

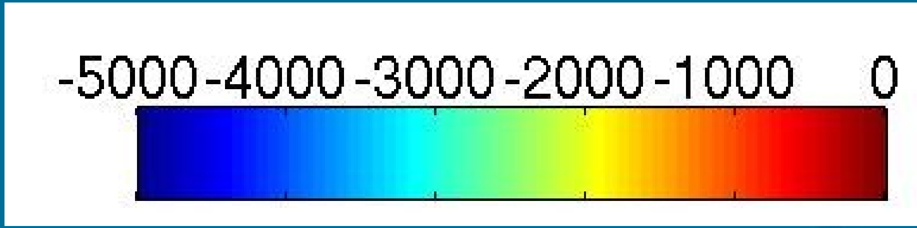


Depth (0.1)



Depth + Mask



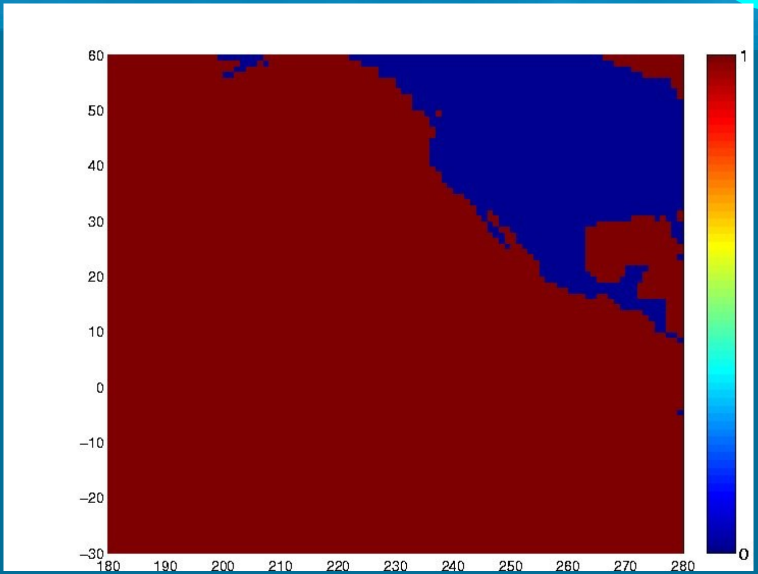


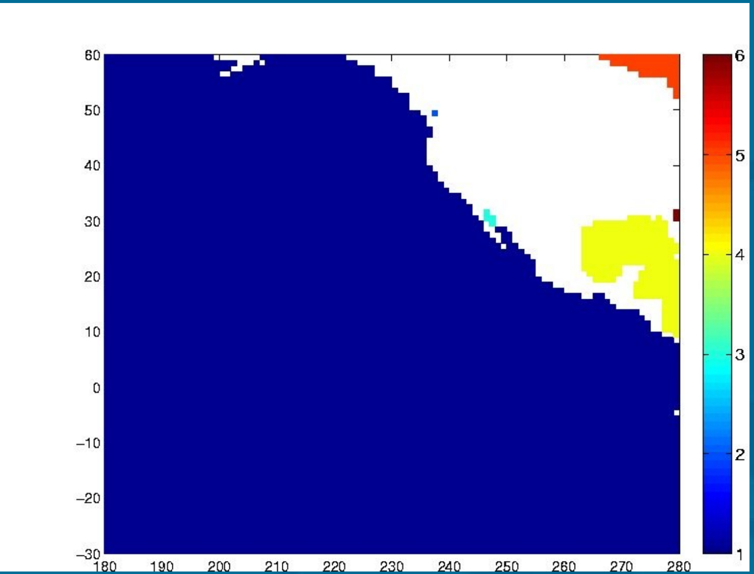
Wet cell clean up routine

- Cycle through all the wet cells and flag all connected cells with the same id
- Independent water bodies have different ids
- Function returns an id map that allows users to switch cells of a particular water body from wet to dry
- Switching of cells can either be done inside the routine with a flag option, or outside by the user

Wet Cell clean up (contd.)

- Initialize all wet cells as unmarked
- Starting from first unmarked cell with marker value at 1, mark all connected wet cells with the same marker
- If more unmarked cells then increment marker by 1 and repeat step 2.
- Keep repeating steps 3 and 2 till no longer unmarked wet cells
- End result is a mask map with the wet cells grouped into independent water bodies





Obstruction grid – Proof of concept

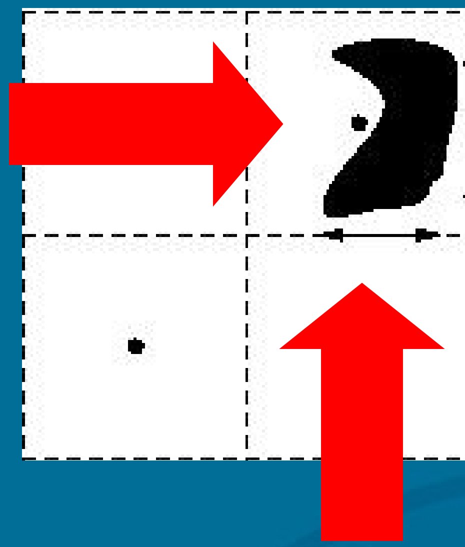
Tolman (2003) showed that sub-grid islands can be modeled in WAVEWATCHIII by physically reducing the energy fluxes between the cells

1D Spatial propagation in WAVEWATCHIII

$$F_i^{n+1} = F_i^n + \frac{\Delta t}{\Delta x}\left(\alpha_{i,-} G_{i,-} - \alpha_{i,+} G_{i,+}\right)$$

Spectral density

Density flux and transparencies at cell boundaries



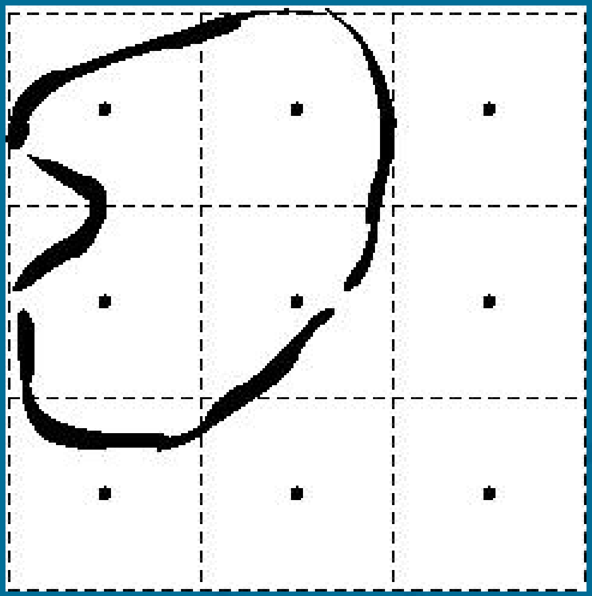
Reduction of energy dependent upon the proportion of cell being obstructed

Obstruction grid ranges from 0 (no obstruction) to 1 (full obstruction)

Two obstruction grids (for the 2 directions of motion) used in WAVEWATCHIII

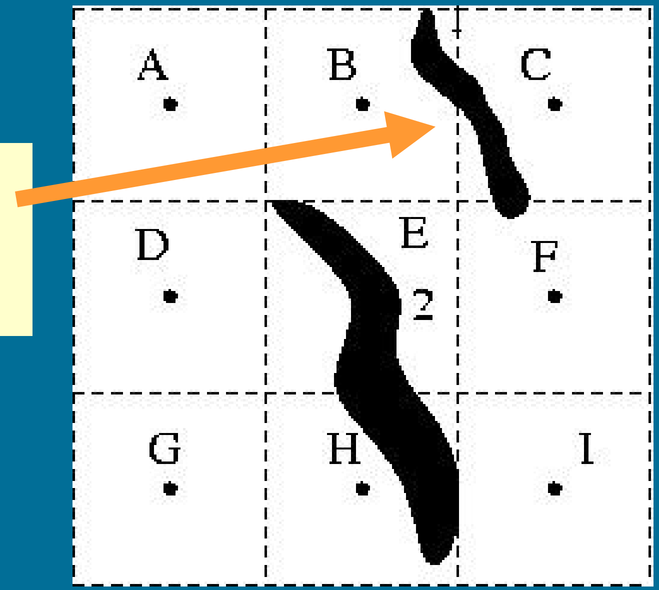
Building an obstruction grid

- Boundary polygons ideal for building obstruction grids
- Obstruction computed as *proportion* of cell length obstructed by boundary (ies)
- Obstruction data for cells next to dry cells set to 0 (to avoid spurious energy decay)
- Sx = obstruction along x = obstruction height/cell height
- Sy = obstruction along y = obstruction width/cell width



(a) Boundaries crossing cells in the same path

Energy flux from B to C should be fully obstructed



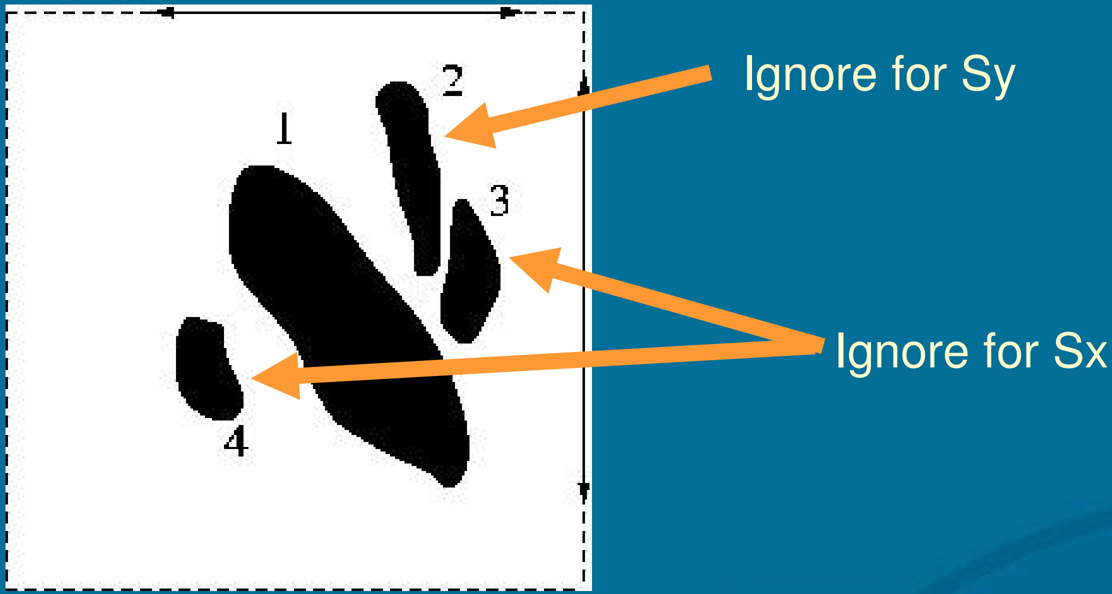
Option1: Account for obstruction path in neighboring cells

Option2: Move boundary segments from common boundary in neighboring cells to the same cell

Using option 2 prevents over counting

Points to consider while building an obstruction grid (contd.)

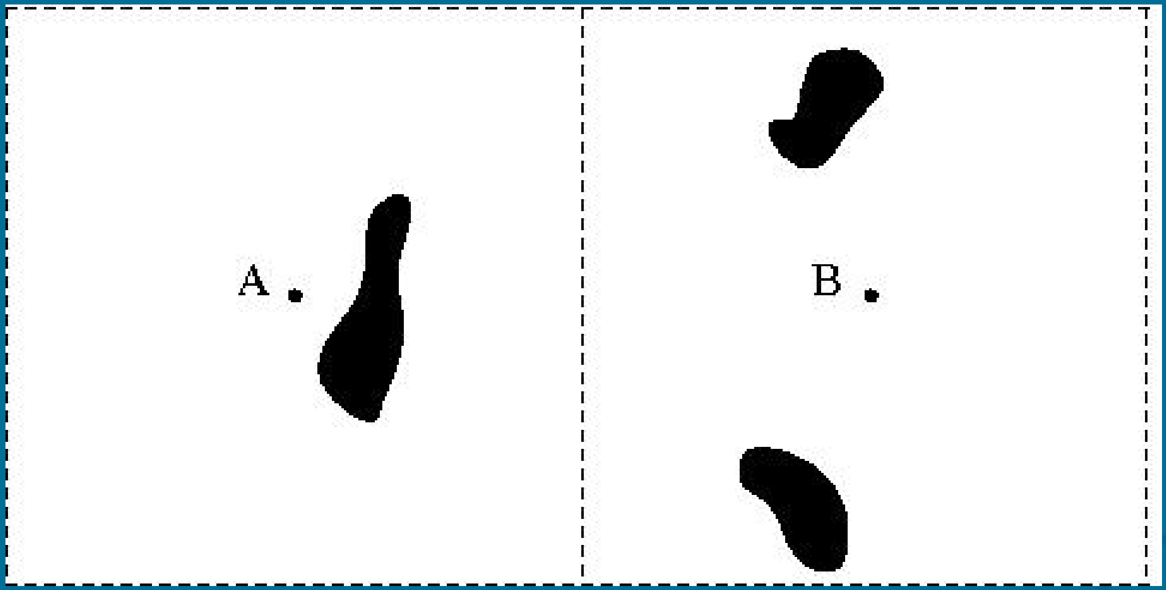
(b) Multiple boundaries within a cell



Obstruction should not be determined from the sum of all lengths but the ***net length***

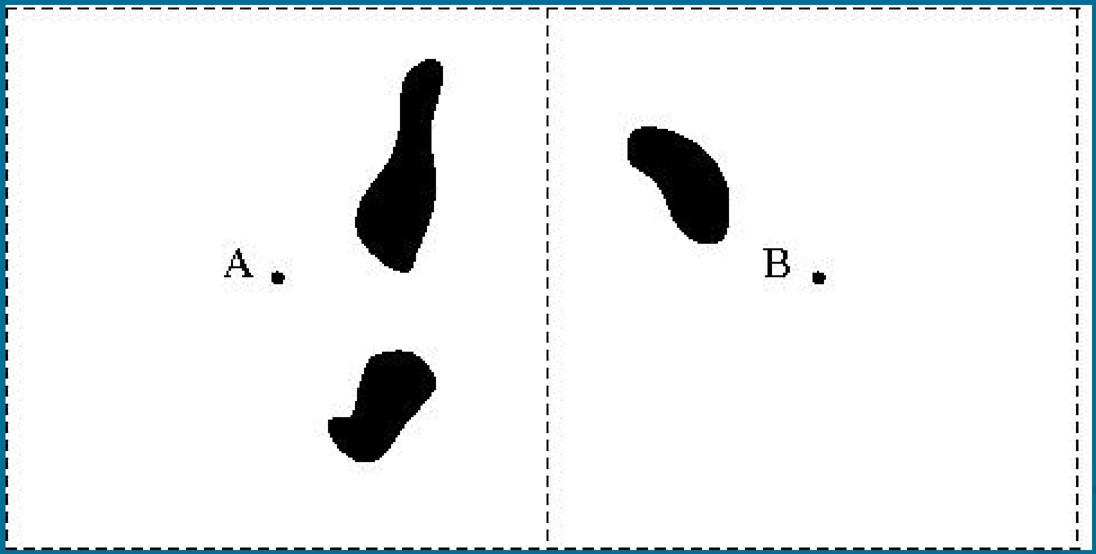
Points to consider while building an Obstruction grid (contd.)

(c) Neighboring cell information



Orientation of boundaries in neighboring cell can lead to greater obstruction than from using boundary information in individual cells only

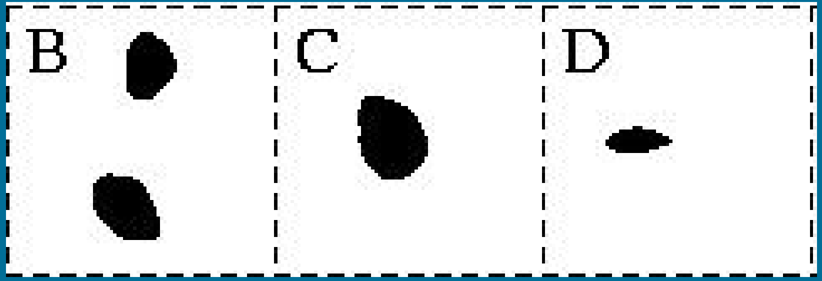
(d) Discount overlapping boundaries from neighboring cells



*Non – zero Sx,Sy values for any particular cell should be computed **if** obstructions in the cell contribute to the obstruction process*

(e) How do you account for neighboring cells ?

Option1: Consider neighbors on both sides



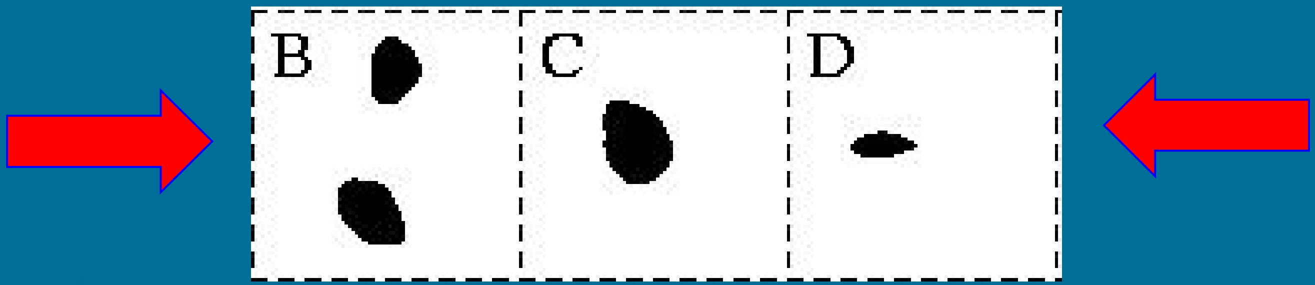
Cell B Sx values would include information from cell C

Cell C Sx values would include information from cell B

Wave propagation from left to right (or right to left) will lead to over attenuation

(e) How do you account for neighboring cells (contd.)?

Option2: Consider neighbors on one side alone



Cell B Sx values would include information from cell C (neighbor to right)

Cell C Sx values would include information from cell B (neighbor to left)

Use right neighbor for wave propagation from right to left

Use left neighbor for wave propagation from left to right

Numerical Tests to validate obstruction algorithm

- 3 different regions
  - Caribbean Islands
  - Hawaii
  - French Polynesian Islands
- For each region
  - 5 grid resolutions (2’, 4’, 8’, 15’ and 30’)
  - 4 different scenarios
    - No obstruction
    - Obstruction grids based on individual cell info only
    - Obstruction grids based on cell info from one neighbor
    - Obstruction grids based on cell info from both neighbors

- Constant swell applied along Northern and Eastern boundaries
  - Hs = 4m, Tp = 10 sec
  - Swell direction = 45^{o} from the North East
  - Directional spread = 20^{o}
  - Monochromatic frequency component
  - 72 directional components (to minimize Garden Sprinkler Effects)
- Tests limited to swell propagation
  - No refraction
  - Source terms switched off

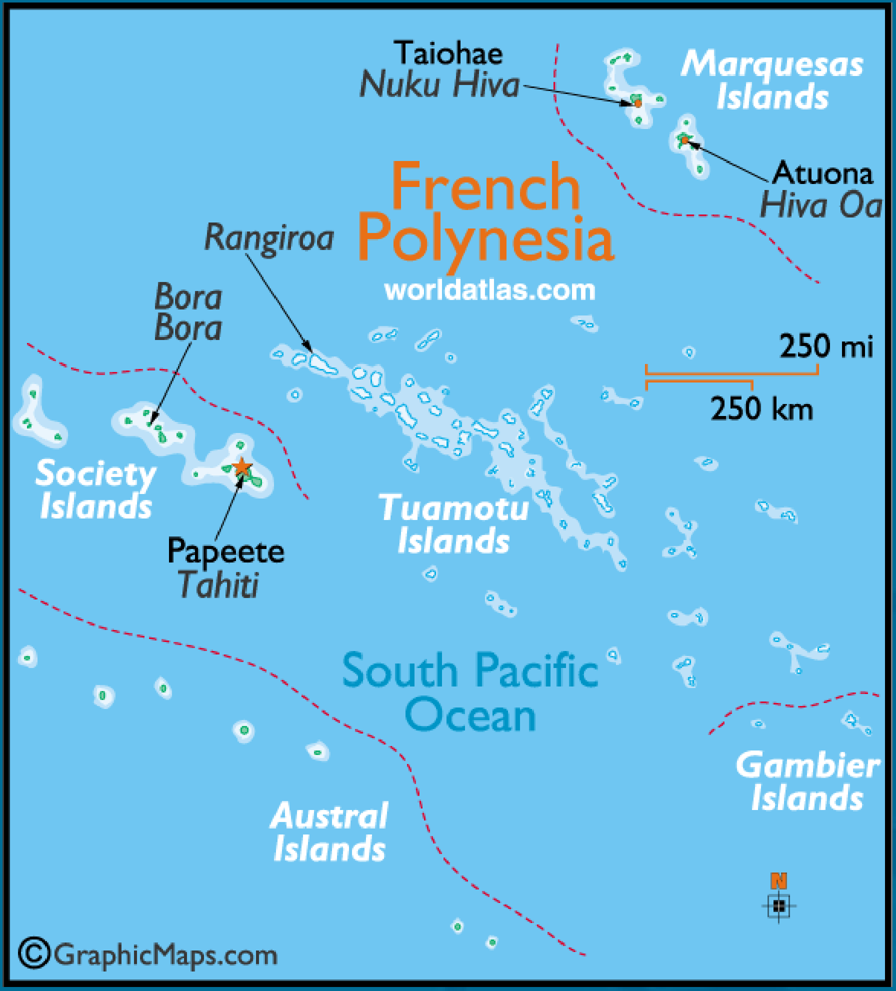
Test Case – French Polynesia

Total # of boundary polygons for this region = 1640

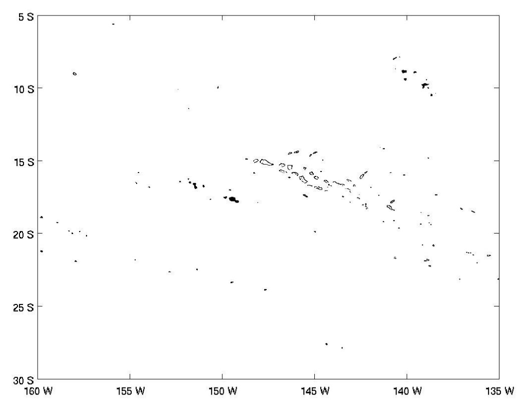
Max projected area ~ 2400 km^2

Min projected area ~ 0.0092 km^2

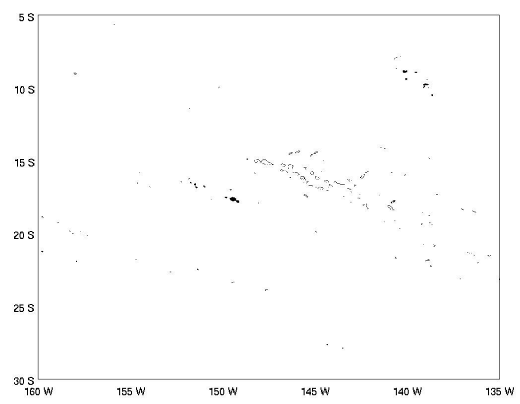
Projected area = length*width



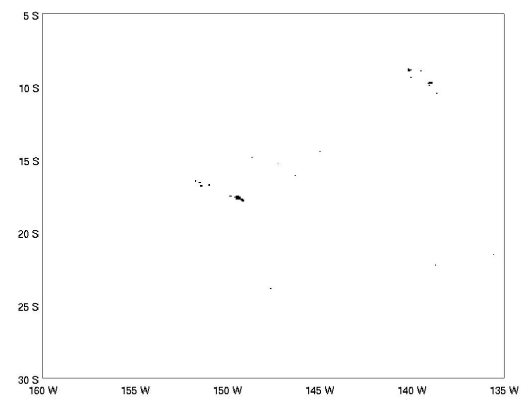
Grids (land – sea masks)



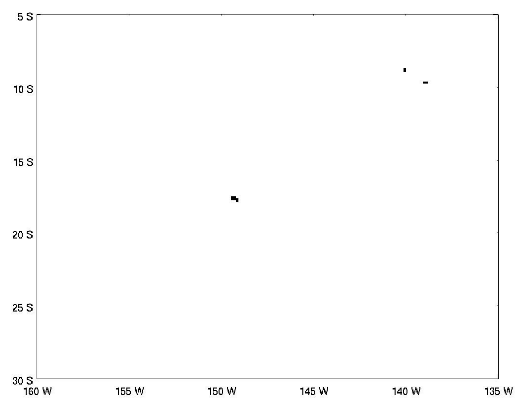
GSHHS



2’ grid



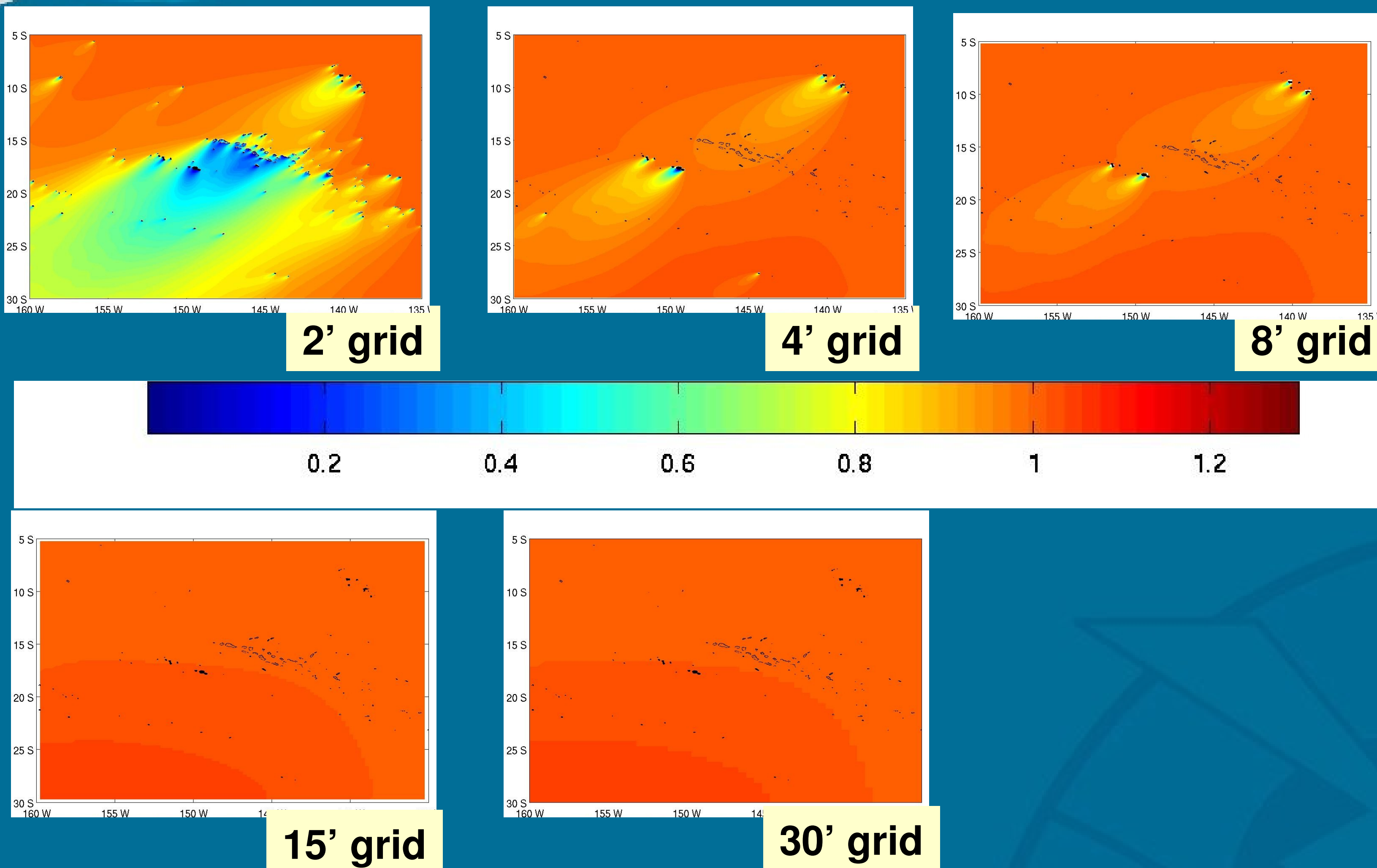
4’ grid



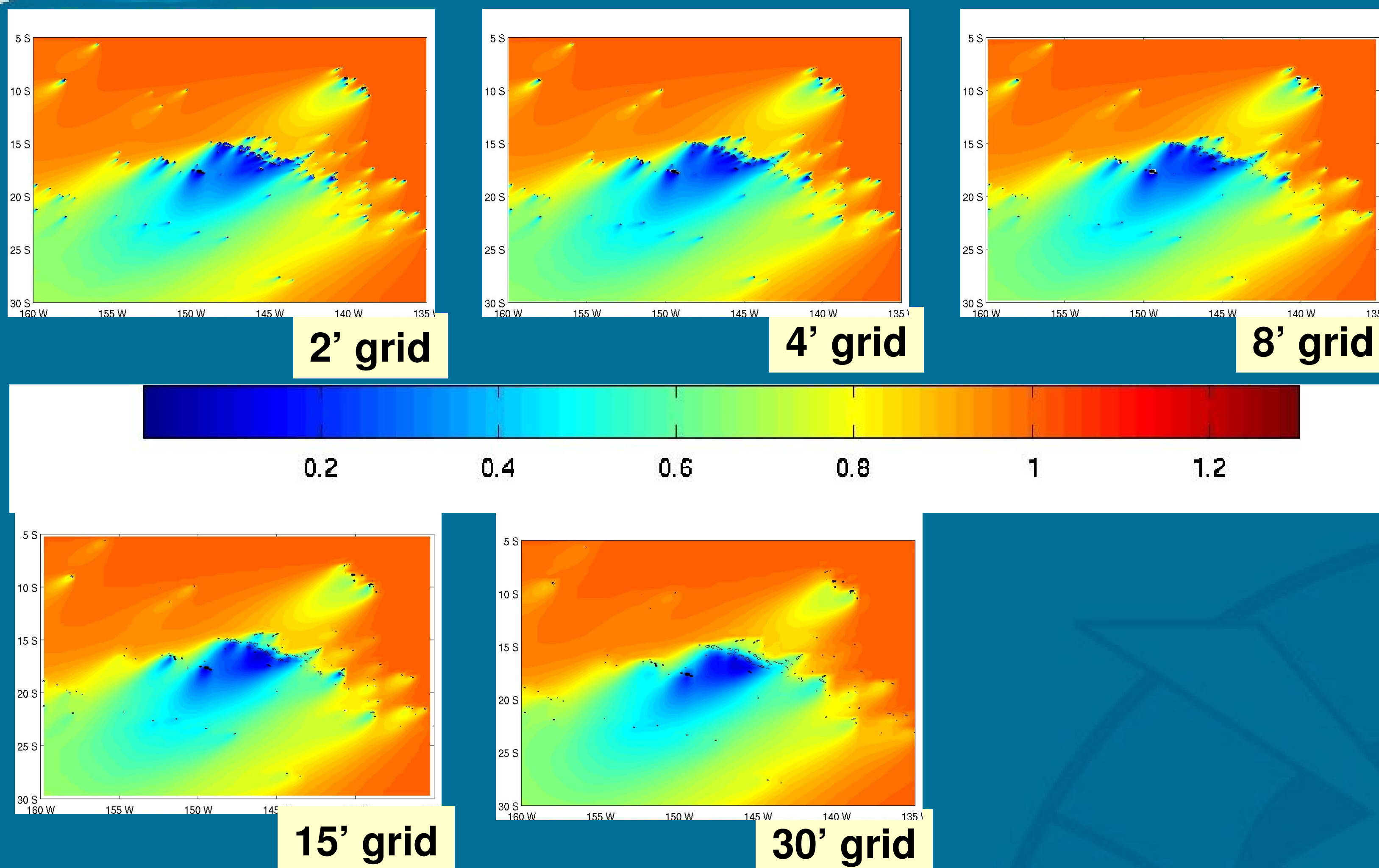
8’ grid

Coarser grids had no land boundaries

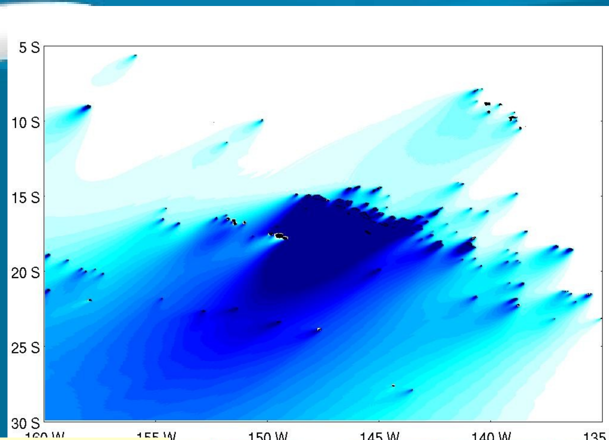
Swell propagation without obstruction grids



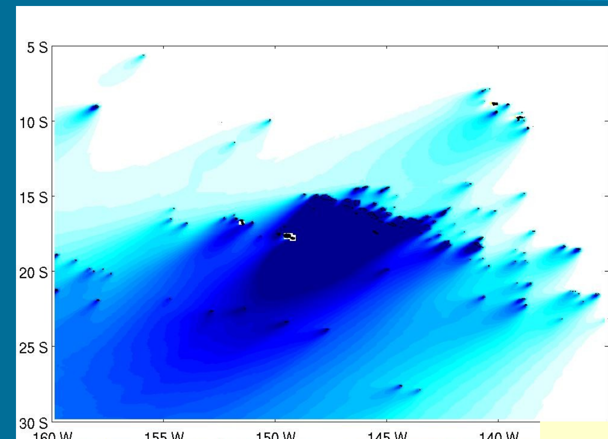
Swell propagation with obstruction grids



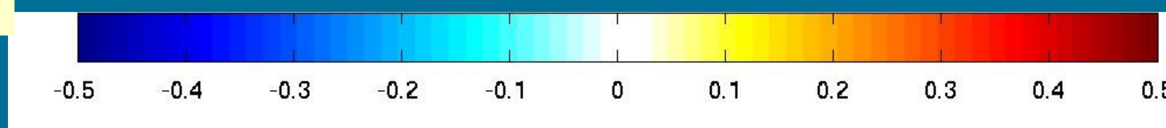
Difference plots (no obstruction)

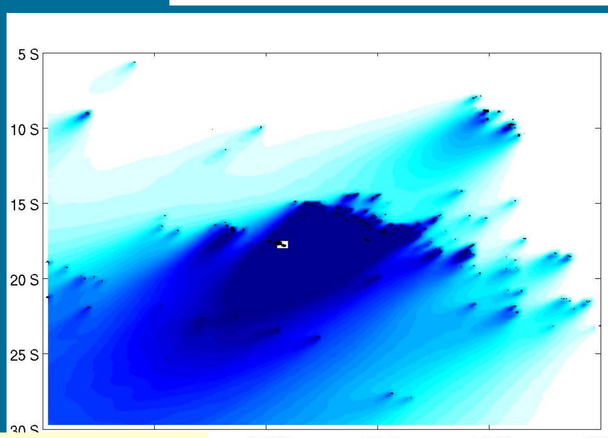


(a) 2’ – 4’

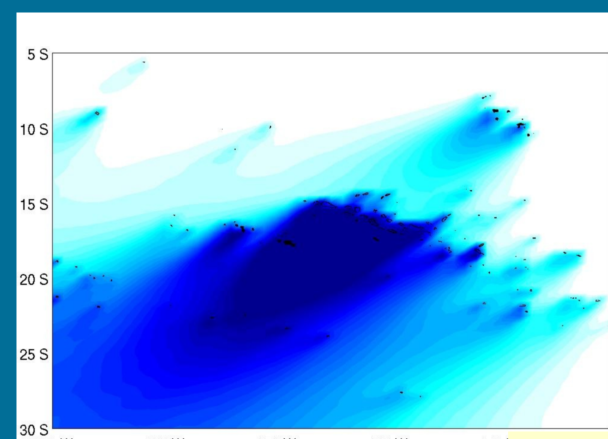


(b) 2’ – 8’



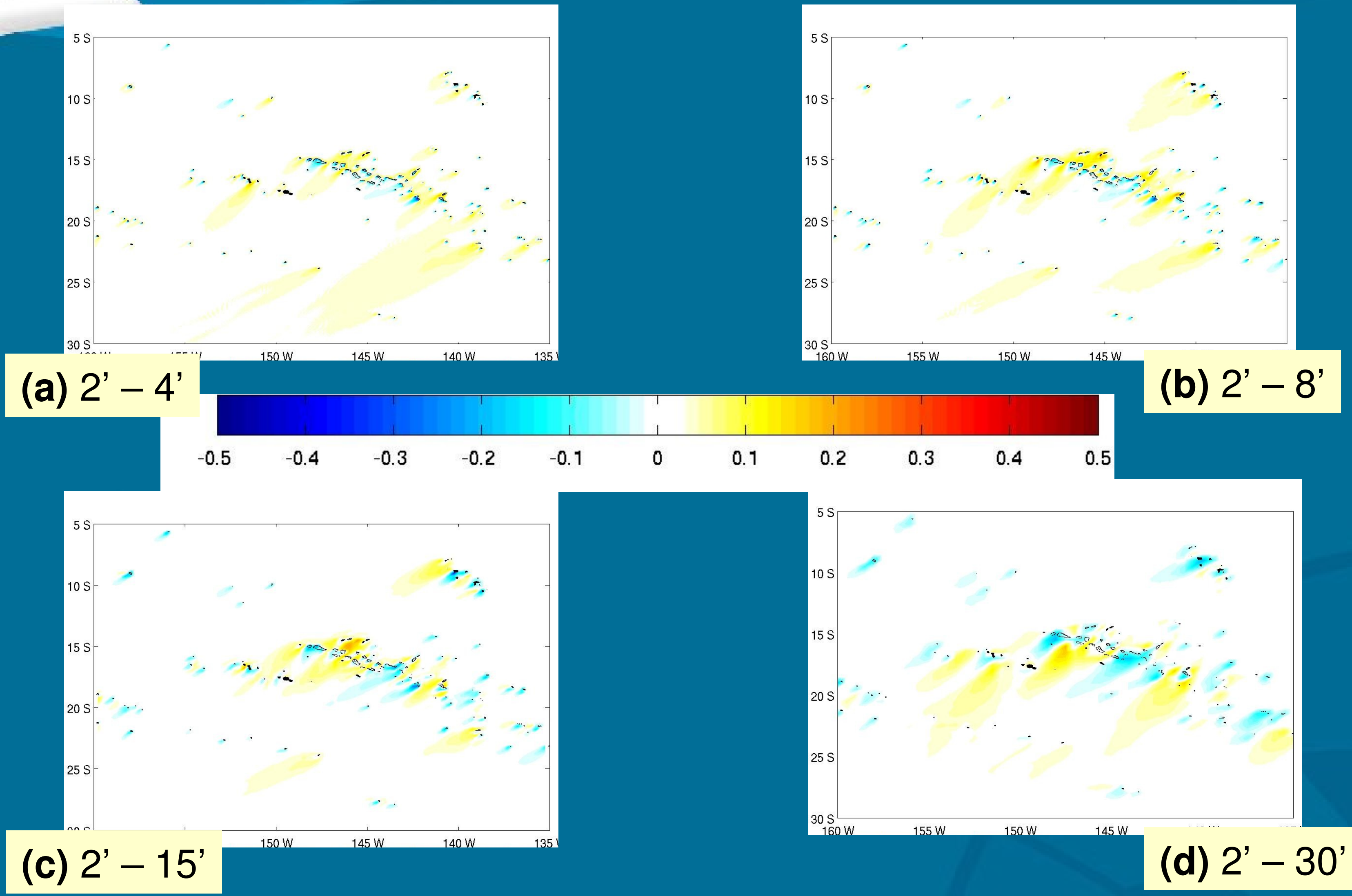


(c) 2’ – 15’

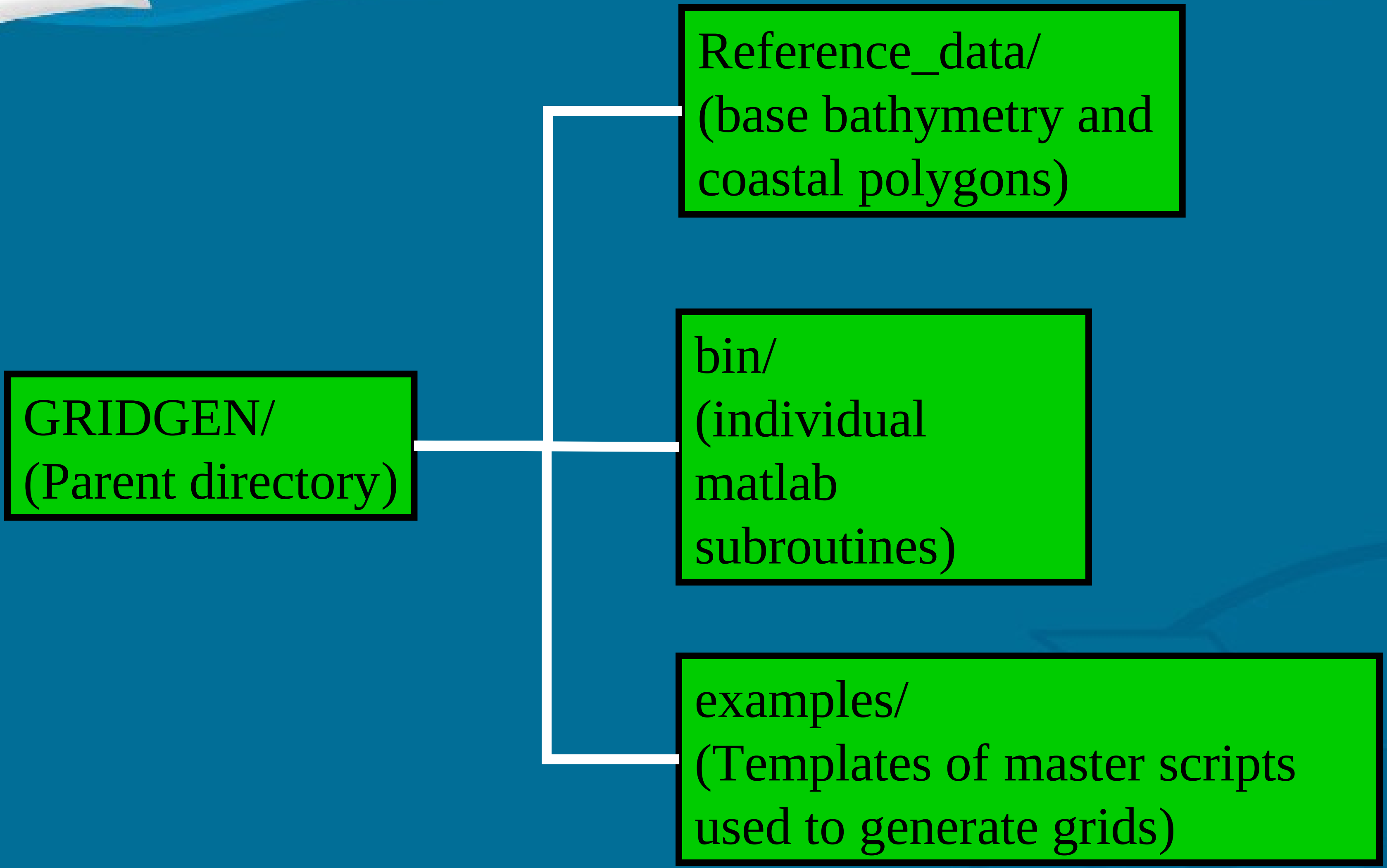


(d) 2’ – 30’

Difference plots (with obstruction)



Putting it together



Example Script for generating a grid

```
% THIS IS AN EXAMPLE SCRIPT FOR GENERATING A REGIONAL GRID AND CAN BE USED AS
A TEMPLATE FOR
% DESIGNING GRIDS

% 0. Initialization

% 0.a Path to directories

  bin_dir = '/export/lnx375/wd20ac/matlab/svn_gridgen/gridgen/bin';
  ref_dir = '/export/lnx375/wd20ac/matlab/svn_gridgen/gridgen/reference_data';
  out_dir = '/export/lnx375/wd20ac/grids_for_india/take_2';

% 0.b Design grid parameters

  fname_poly = 'user_polygons.flag';
  fname = 'ind_ocean';
  icoords = 1;  % longitude range  (0 -> -180 -180; 1 -> 0 - 360 )
  grid_box = [-33 24 32 125];  % starting and ending lat,lon for grid domain
  dx = 0.5;                    % grid resolution in x (degrees)
  dy = 0.5;                    % grid resolution in y (degrees)
  ref_grid = 'etopo1';         % reference grid source
  boundary = 'full';           % boundary option (which GSHHS file to load)
```

Define directories

Grid Parameters

Example script for generating grid (contd).

Set paths and boundary flags

```
% 0.c Setting the paths for subroutines

  addpath(bin_dir,'-END');

% 0.d Reading Input data

  read_boundary = 1;                 % flag for reading boundary info
  opt_poly = 1;                      % flag for reading user defined polygons.

  if (read_boundary == 1)

      fprintf(1,'.........Reading Boundaries..................\n');

      load([ref_dir,'/coastal_bound_',boundary,'.mat']);

      N = length(bound);
      if (icoords == 0)
          for i = 1:N
              loc = find(bound(i).x > 180);
              bound(i).x(loc) = bound(i).x(loc) - 360;
              bound(i).west = min(bound(i).x);
              bound(i).east = max(bound(i).x);
              clear loc;
          end;
```

Load boundary polygons

Example script for generating grid (contd).

```
   elseif (icoords == 1)
         for i = 1:N
             loc = find(bound(i).x < 0);
             bound(i).x(loc) = bound(i).x(loc) + 360;
             bound(i).west = min(bound(i).x);
             bound(i).east = max(bound(i).x);
             clear loc;
         end;
     end;

     if (opt_poly == 1)
         [bound_user,Nu] = optional_bound(ref_dir,...
                                fname_poly,icoords);
     end;
     if (Nu == 0)
         opt_poly = 0;
     end;

 end;
```

Example script for generating grid (contd).

Generate Bathymetry grid

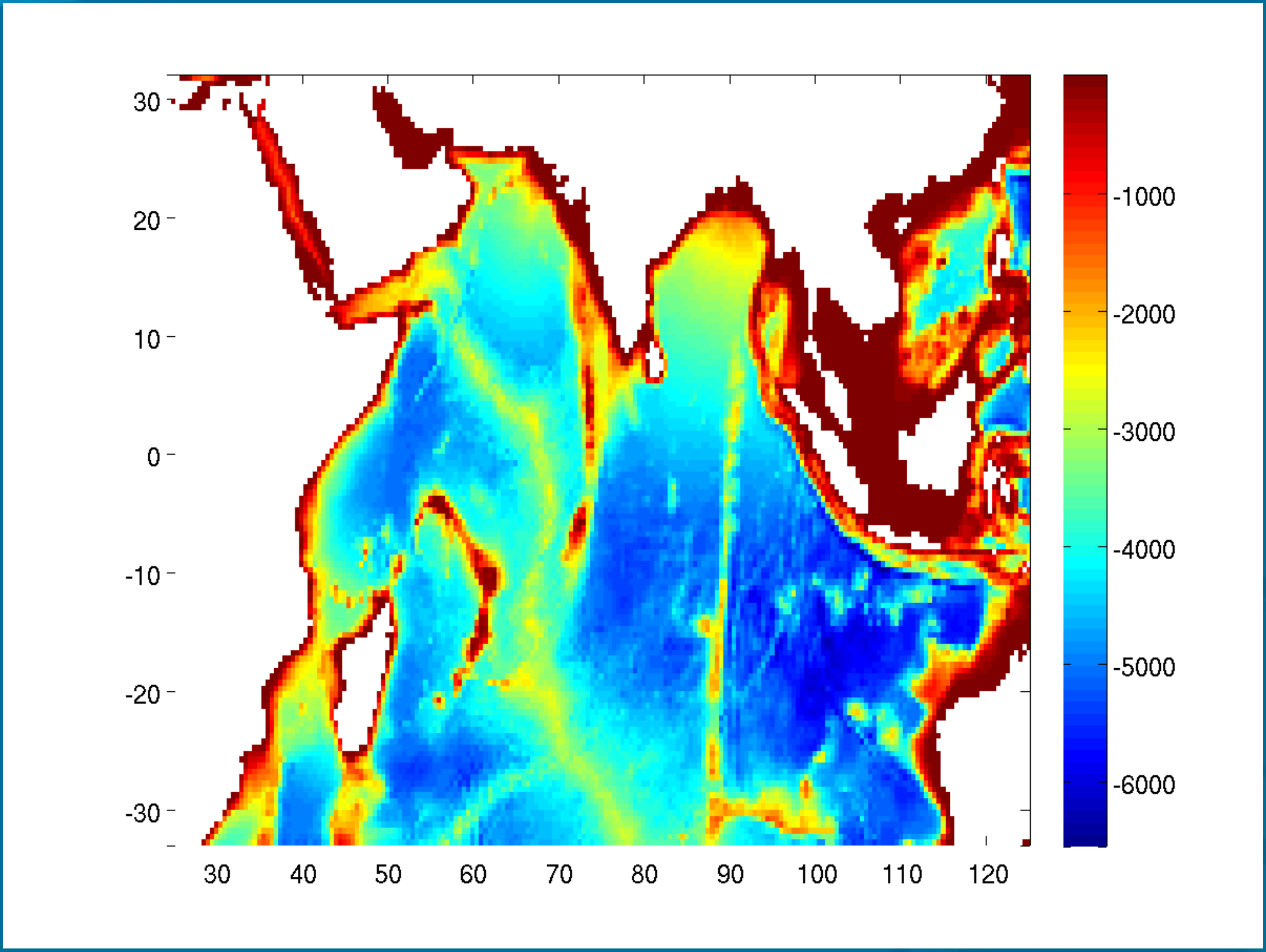
```
% 1. Generate the grid

  fprintf(1,'........Creating Bathymetry.................\n');

  [lon,lat,depth] = generate_grid(ref_dir,ref_grid,grid_box,...
                                dx,dy,icoords,0.1,0,999);
```

Ratio determining wet cell / dry cell cut off

Water depth separating wet cell from dry cells

Default value for dry cells

Initial Bathymetry



Example script for generating grid (contd).

```
% 2. Computing boundaries within the domain

% 2.a Set the domain big enough to include the cells along the edges of the
grid

  lon_start = lon(1)-dx;
  lon_end = lon(end)+dx;
  lat_start = lat(1)-dy;
  lat_end = lat(end)+dy;

  coord_start = -180+icoords*180;
  coord_end = coord_start + 360;

  if (lon_start < coord_start)
      lon_start = lon_start + 360;
  end;
  if (lon_end > coord_end)
      lon_end = lon_end - 360;
  end;
```

Setting up box for extracting boundaries

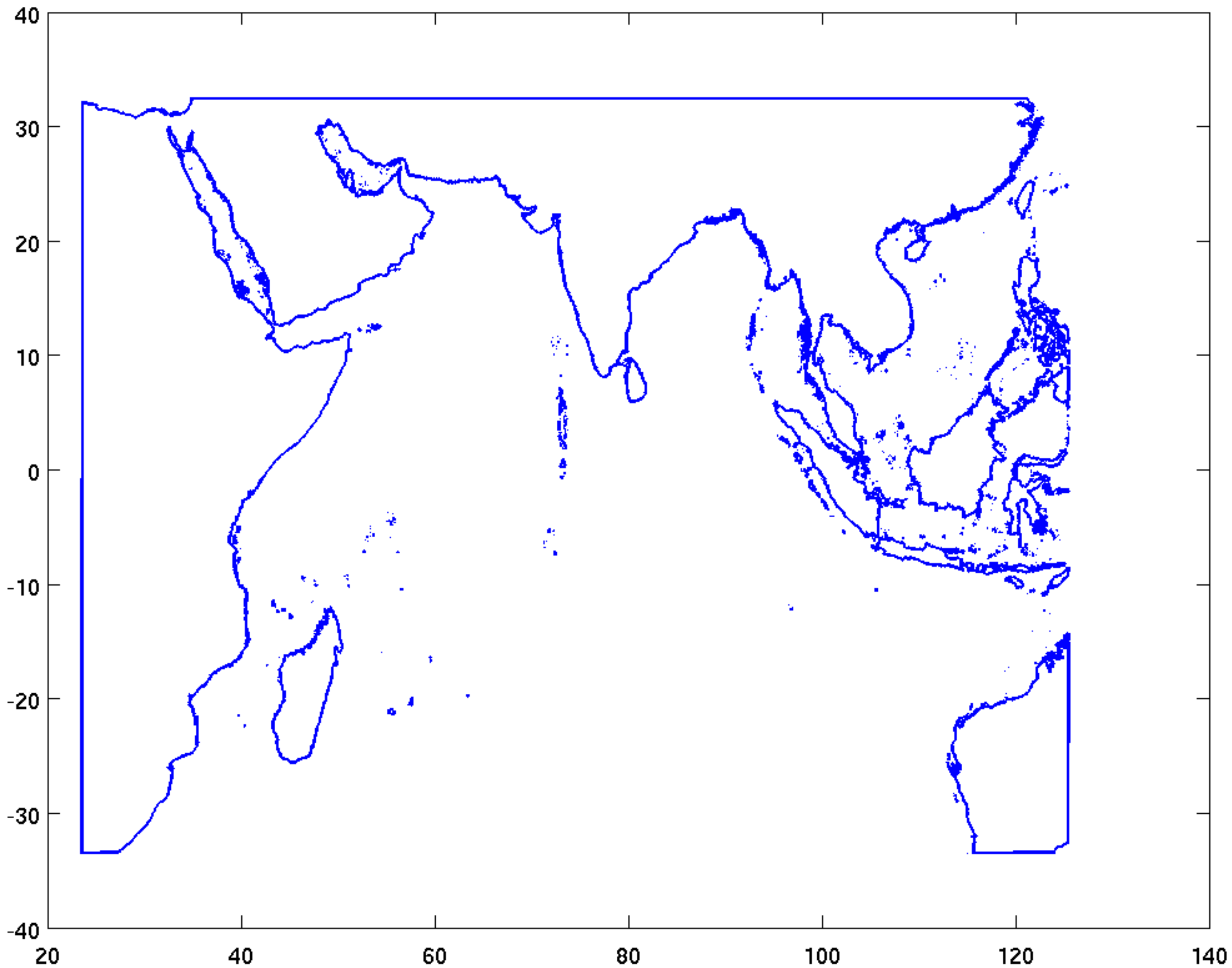
Example script for generating grid (contd).

```
% 2.b Extract the boundaries from the GSHHS and the optional databases

  if (lon_start < lon_end)

      coord = [lat_start lon_start lat_end lon_end];
      [b,N1] = compute_boundary(coord,bound);

      if (opt_poly == 1)
          [b_opt,N2] = compute_boundary(coord,bound_user);
      end;

  end;
```

Boundary extraction routine

Domain for extracting boundaries

Reference boundaries



40
30
20
10
0
-10
-20
-30
-40
20
40
60
80
100
120
140

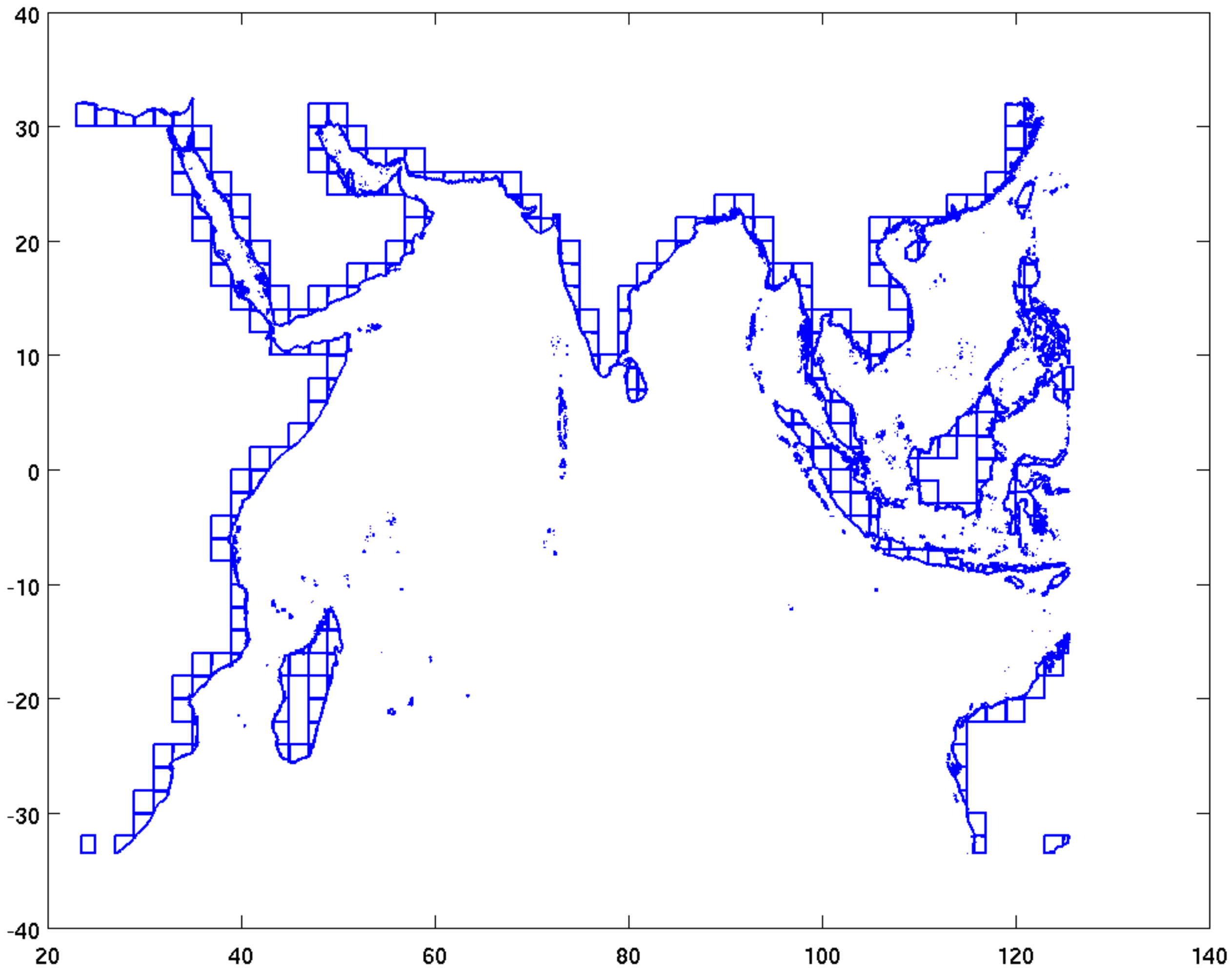
Example script for generating grid (contd).

```
% 3. Set up Land - Sea Mask

% 3.a Set up initial land sea mask.

  m = ones(size(depth));
  loc = find(depth == 999);
  m(loc) = 0;

% 3.b Split the larger GSHHS polygons for efficient computation of the land sea
mask. (Optional)
  fprintf(1,'.........Splitting Boundaries..

  b_split = split_boundary(b,2);

% 3.c Get a better estimate of the land sea 

  fprintf(1,'.........Cleaning Mask........

  m2 = clean_mask(lon,lat,icoords,m,b_split,0.5);
```

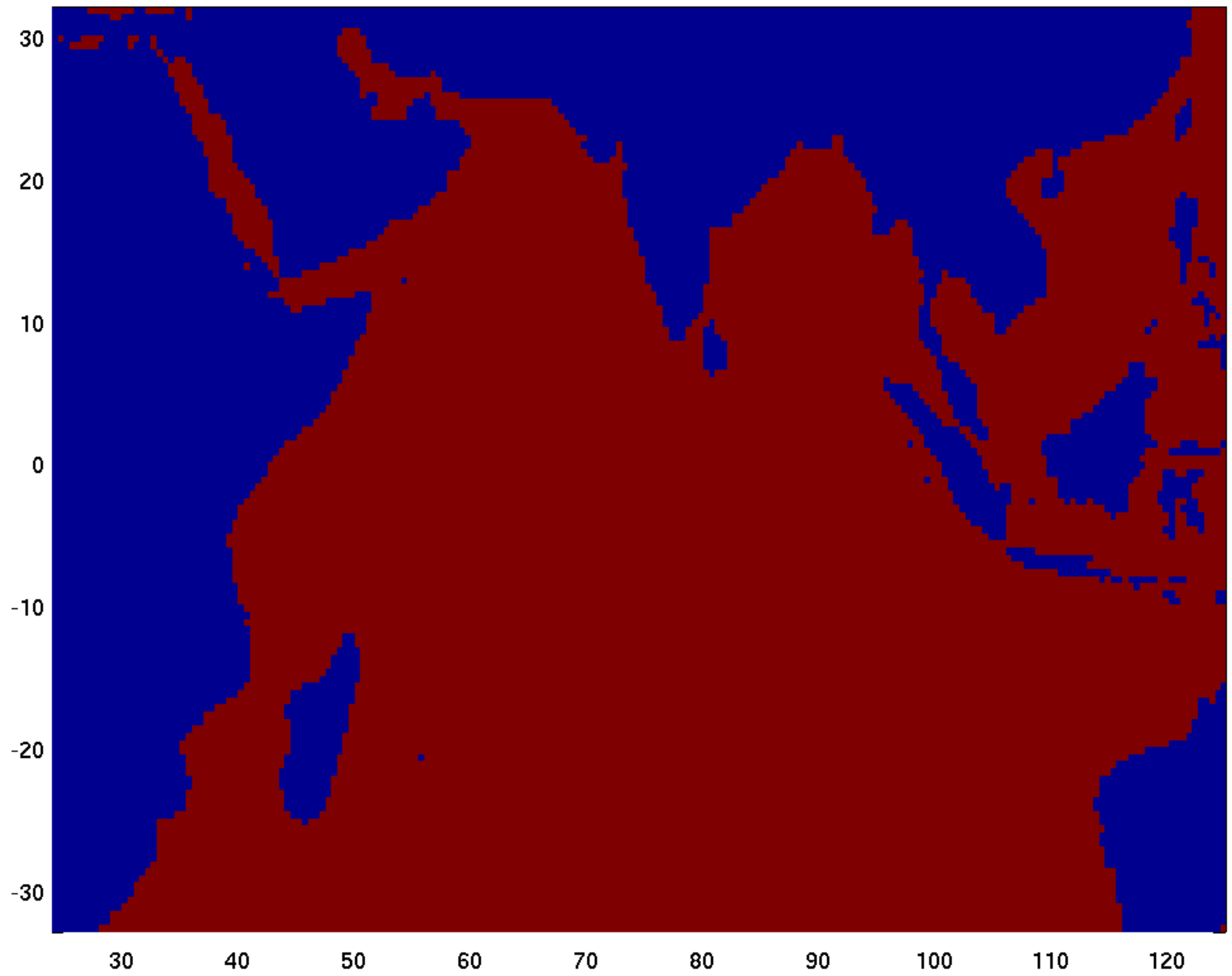
Initial land sea mask

Optional algorithm for splitting boundaries into manageable sizes (*used only for clean_mask routine)*

mask


40
30
20
10
0
-10
-20
-30
-40
20
40
60
80
100
120
140

Land – Sea mask



Example script for generating grid (contd).

```
% 3.d Remove lakes and other minor water bodies

  fprintf(1,'........Separating Water Bodies.................\n');

  [m4,mask_map] = remove_lake(m2,-1,0);
```

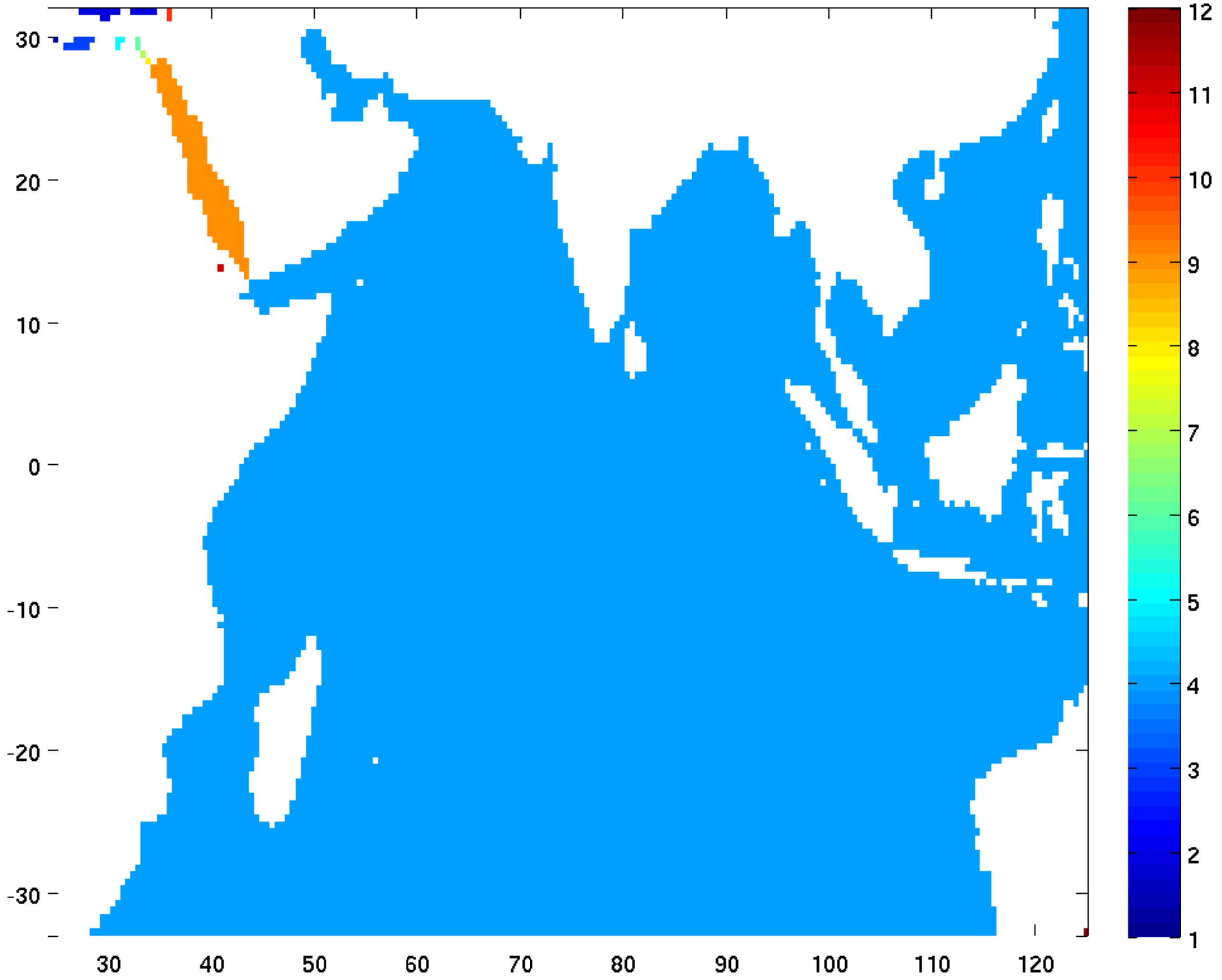
Map of water bodies

Tolerance value (if positive all water bodies having less than this number of wet cells are flagged dry. If negative then all but the largest water body is flagged dry)

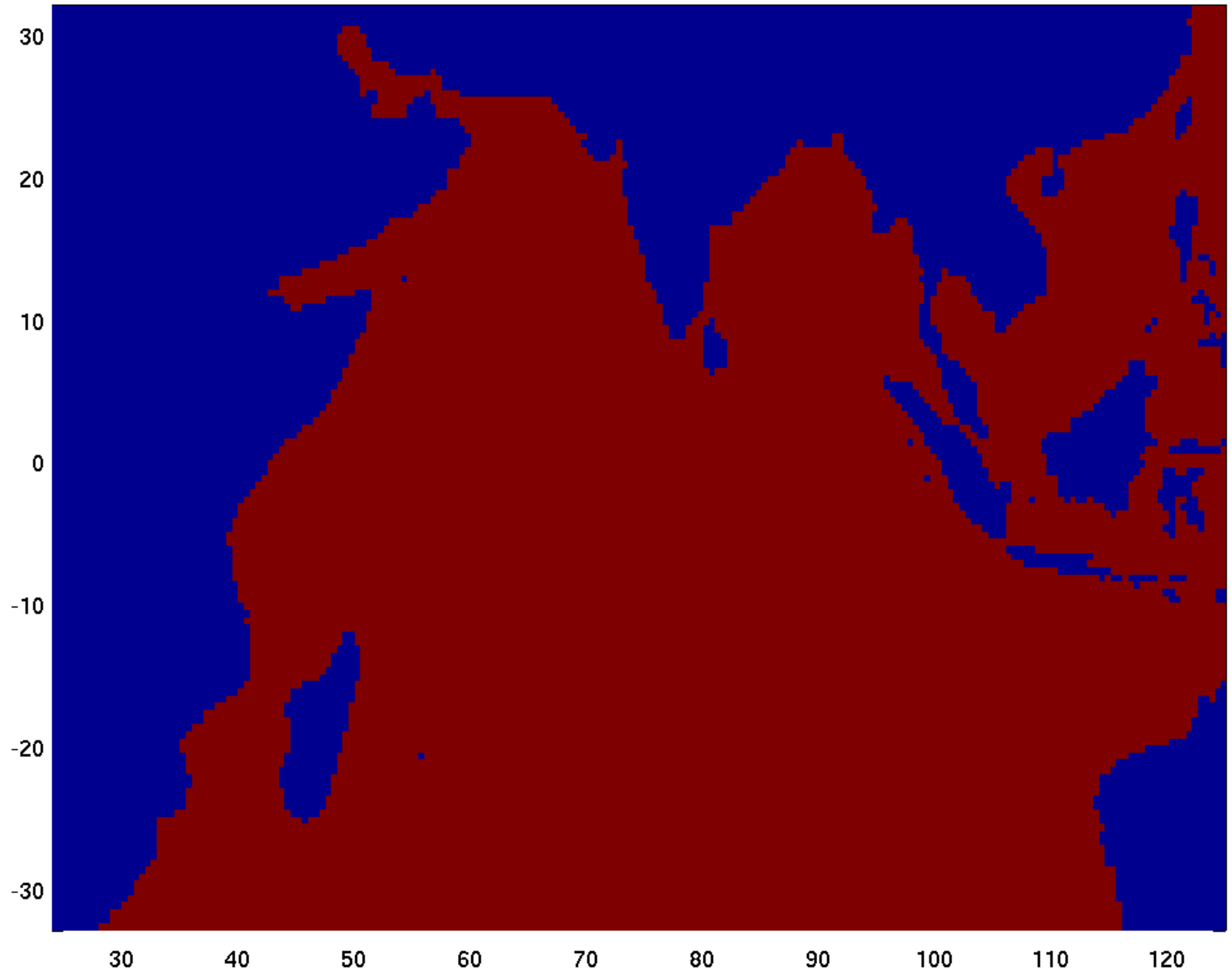
Switch identifying global grid (0 == No, 1 == Yes)

Remove inconsistencies and unnecessary water bodies

Map of connected water bodies



Final Land – Sea mask

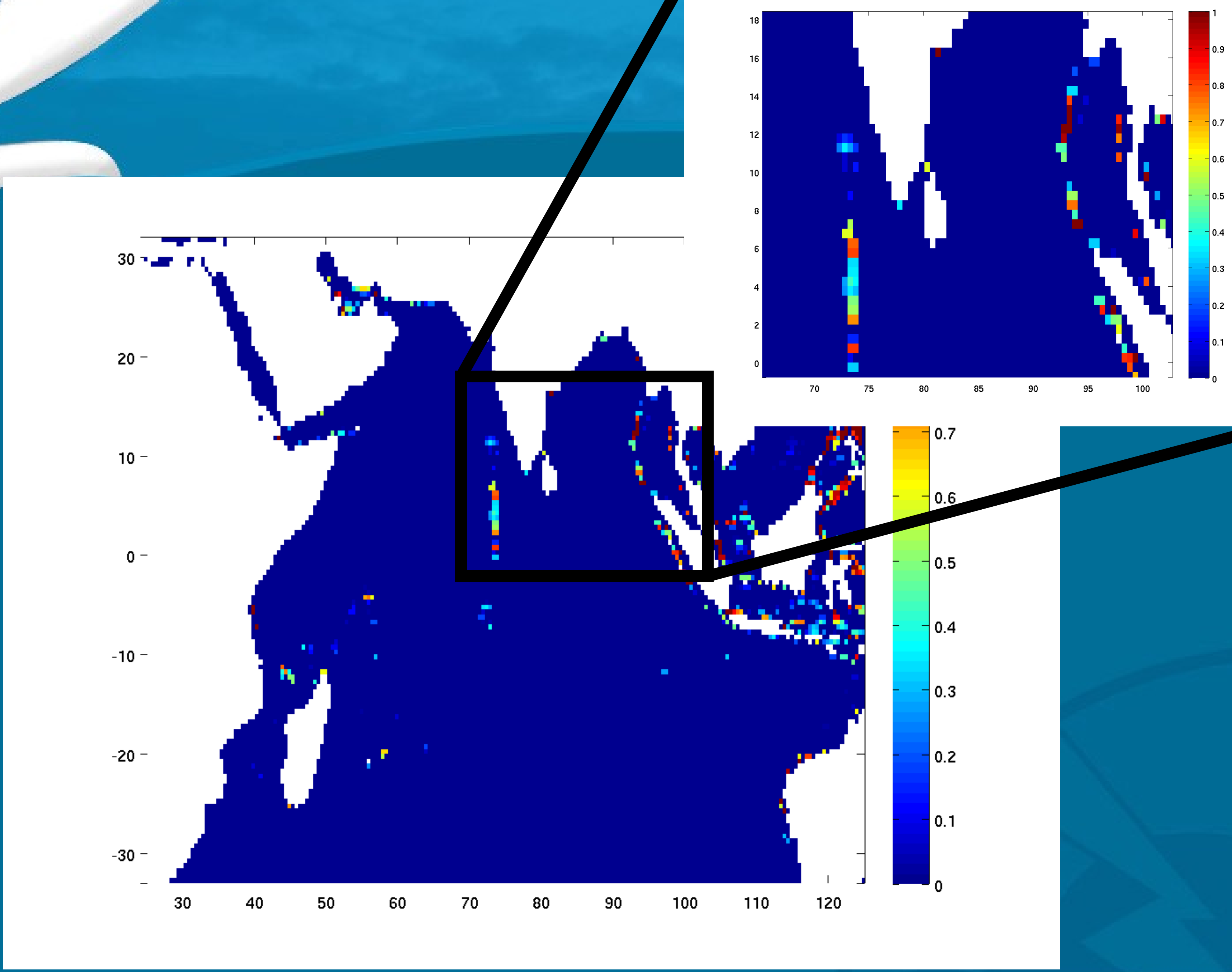


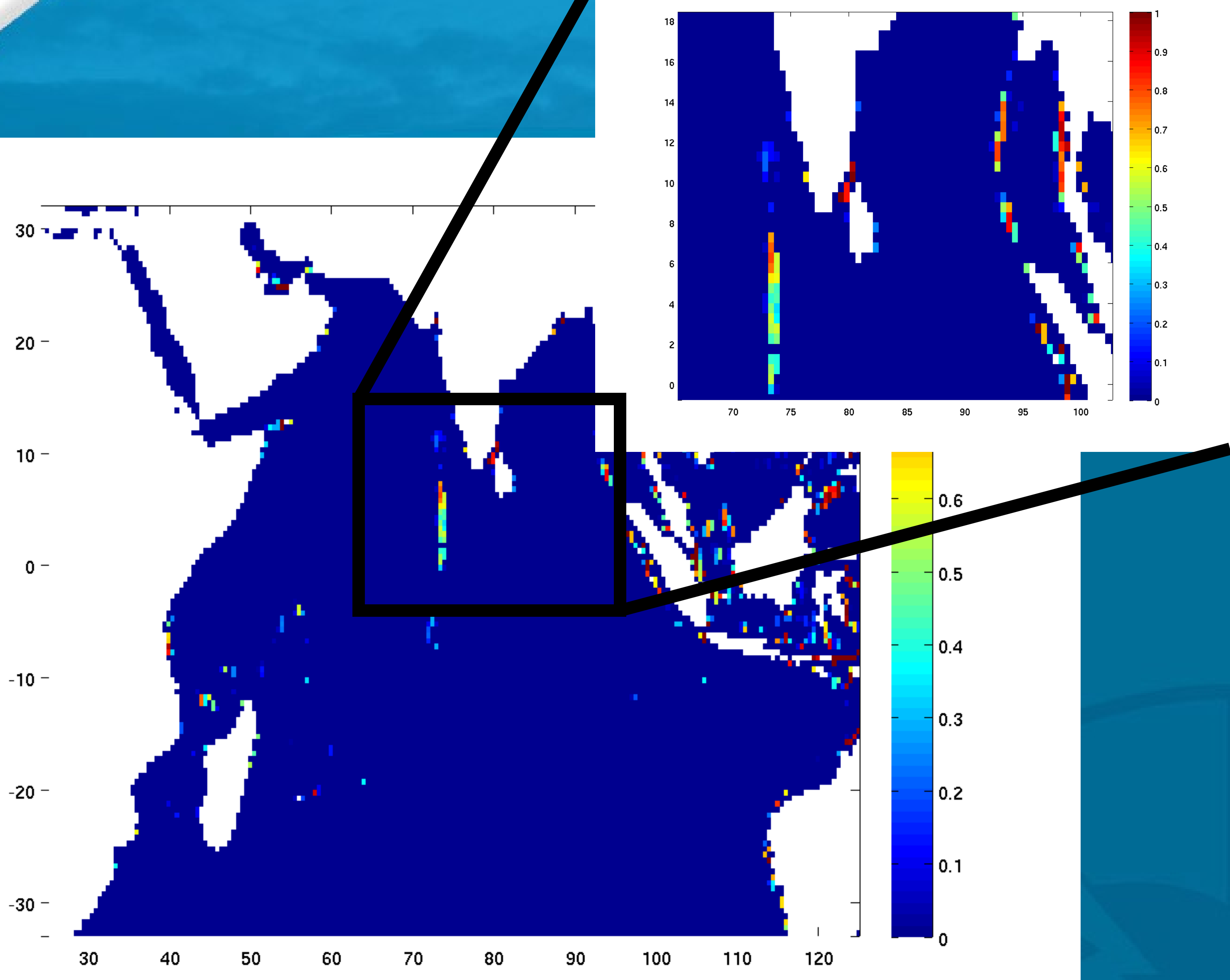
Example script for generating grid (contd).

```
% 4. Generate sub - grid obstruction sets in x and y
final land/sea mask and the coastal boundaries

  fprintf(1,'........Creating Obstructions..................\n');

  [sx1,sy1] = create_obstr(lon,lat,b,m4,icoords,1,1);

% 5. Output to ascii files for WAVEWATCH III

  depth_scale = 1000;
  obstr_scale = 100;

  d = round((depth)*depth_scale);
  write_ww3file([out_dir,'/',fname,'.depth_ascii'],d);
  write_ww3file([out_dir,'/',fname,'.maskorig_ascii'],m4);
  d1 = round((sx1)*obstr_scale);
  d2 = round((sy1)*obstr_scale);
  write_ww3obstr([out_dir,'/',fname,'.obstr_lev1'],d1,d2);
  write_ww3meta([out_dir,'/',fname],lon,lat,1/depth_scale,...
                                          1/obstr_scale);
```

Generate obstruction grids

Write to ascii files (to be used with ww3_grid)


30
20
10
0
-10
-20
-30
30 40 50 60 70 80 90 100 110 120
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
18
16
14
12
10
8
6
4
2
0
70 75 80 85 90 95 100
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0



30
20
10
0
-10
-20
-30
30 40 50 60 70 80 90 100 110 120
0 0.1 0.2 0.3 0.4 0.5 0.6
18 16 14 12 10 8 6 4 2 0
70 75 80 85 90 95 100
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

Assigning boundary points

- The land – sea mask from the grid generation template script only identifies points as land or sea
- Multi-grid WAVEWATCH III needs land, sea and boundary points (where grid information is exchanged)
- Multi – grid WAVETCH III can also assign the boundary points *inside* a grid, thus including a new type (excluded point)
- An additional routine was developed that would separate the land – sea mask into

  Land (value 0)

  Sea (value 1)

  Boundary (value 2)

  Excluded (value 3)
- To execute this script we need the land – sea mask files of the target grid and the base grid with which it will exchange information
- Hence this script is executed *after* all the grids have been generated

Script for modifying mask file

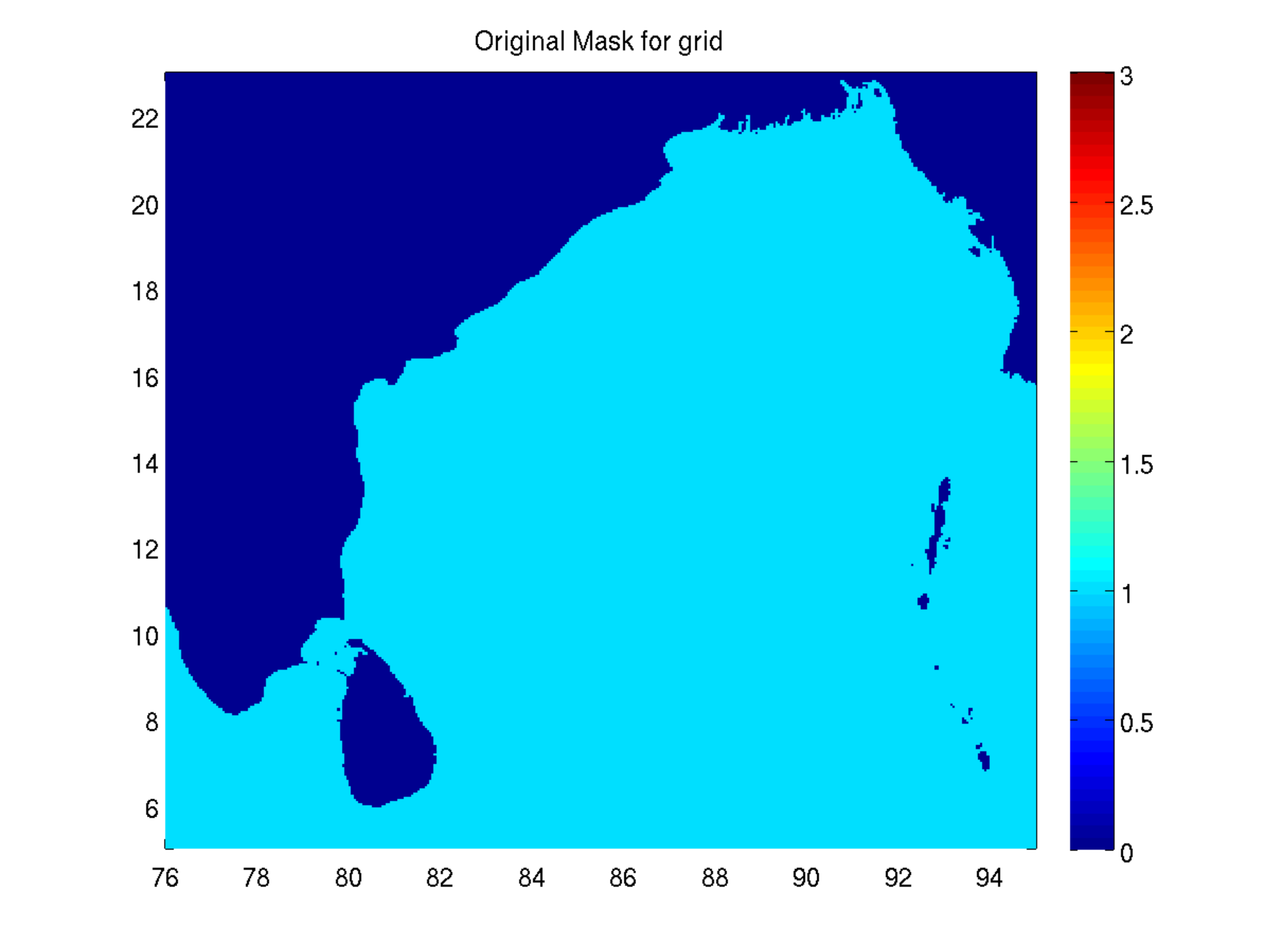
Define paths

Read grid information

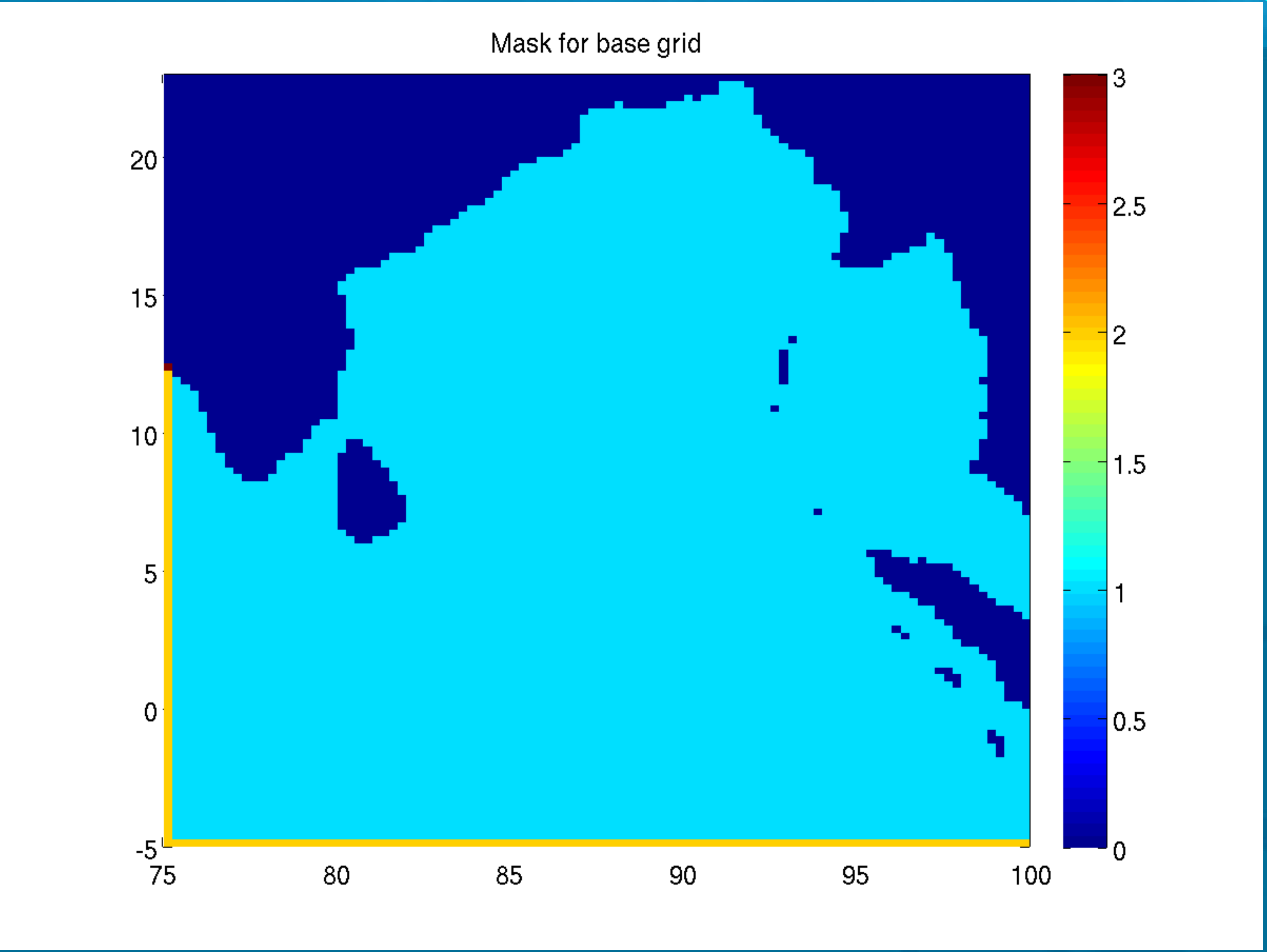
Read base grid information

```
% EXAMPLE SCRIPT FOR MODIFYING MASK FILE FOR MULTI-GRID WW3

 bin_dir
='/export/lnx375/wd20ac/matlab/svn_gridgen/grid
 in_dir  = '/export/lnx375/wd20ac/grids_for_ind
 out_dir = '/export/lnx375/wd20ac/grids_for_ind

 addpath(bin_dir,'-END');

 fname = 'e_coast';
 [lon,lat] = read_ww3meta([in_dir,'/',fname,'.meta']);
 Nx = length(lon);
 Ny = length(lat);
 m = read_mask([in_dir,'/',fname,'.maskorig_ascii'],Nx,Ny);

 fnameb = 'bay_beng';
 [lonb,latb] = read_ww3meta([in_dir,'/',fnameb,'.meta']);
 Nxb = length(lonb);
 Nyb = length(latb);
 mb = read_mask([in_dir,'/',fnameb,'.mask'],Nxb,Nyb);
```

Original mask of grid



Mask of base grid



Script for mod…

Read polygon information (defines the active region)

```
fid = fopen('east_coast.poly_ascii','r');
[a1,count] = fscanf(fid,'%f');
px = a1(1:2:count);
py = a1(2:2:count);
```

```
m_new = modify_mask(m,lon,lat,px,py,mb,lonb,latb,1);
```

Compute new mask

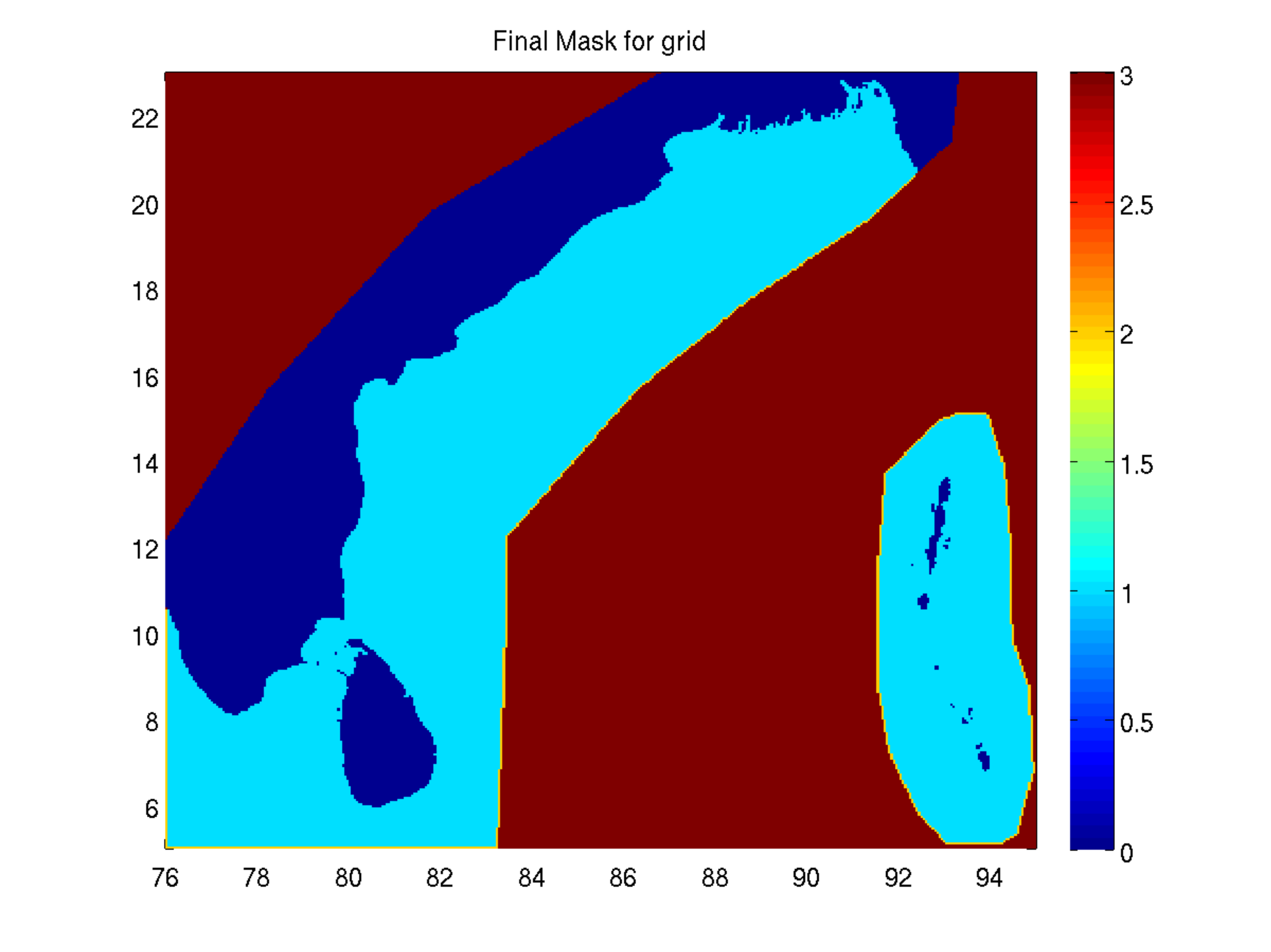
```
fid = fopen('east_coast_anni.poly_ascii','r');
[a1,count] = fscanf(fid,'%f');
px1 = a1(1:2:count);
py1 = a1(2:2:count);
m_tmp = modify_mask(m,lon,lat,px1,py1,mb,lonb,latb,1);

loc = find(m_tmp~=3);
m_new(loc) = m_tmp(loc);
clear loc;
```

Repeat for multiple polygons

```
write_ww3file([out_dir,'/',fname,'.mask'],m_new);
```

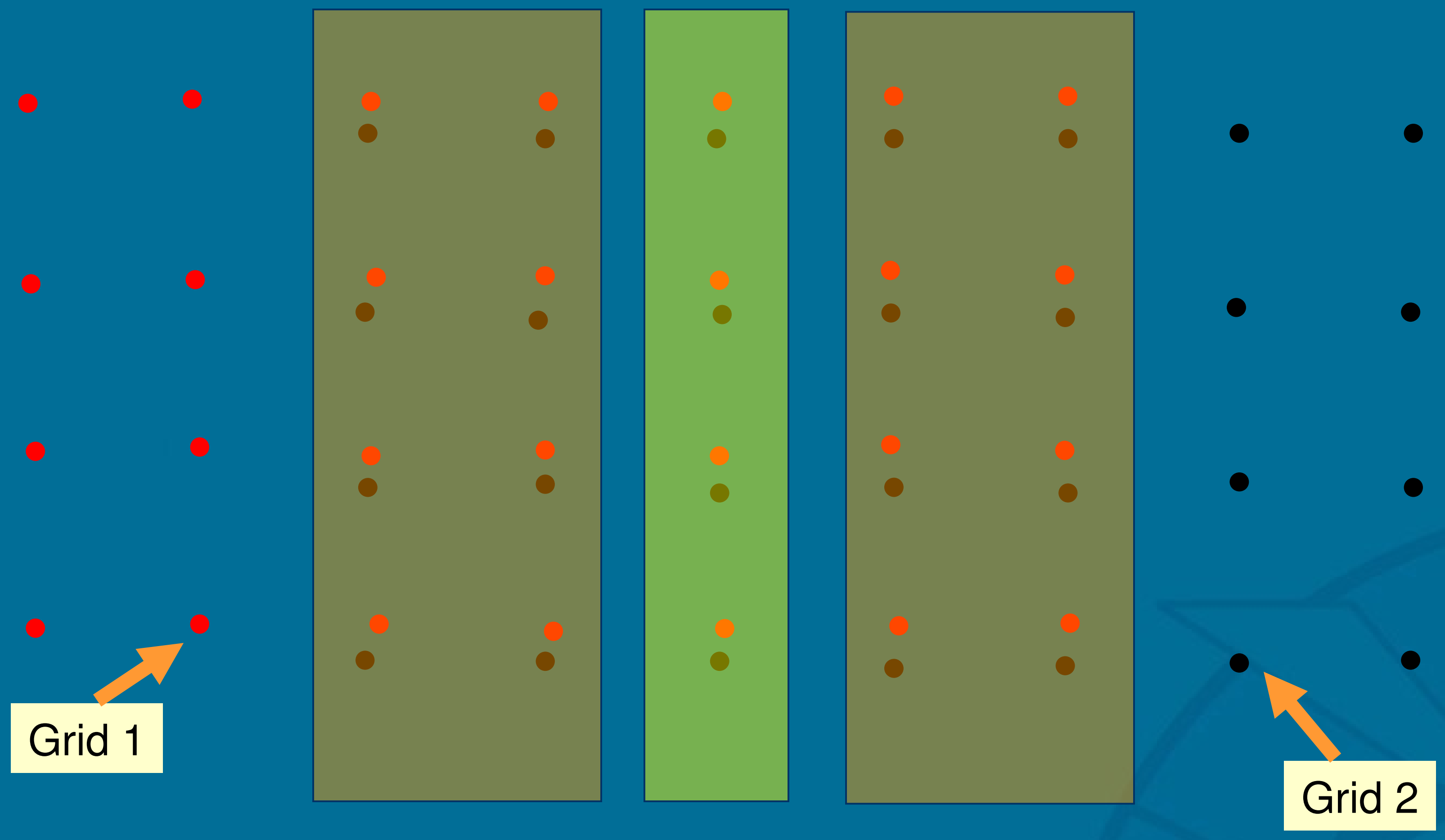
Write new mask to file

Final mask


Final Mask for grid
22
20
18
16
14
12
10
8
6
76
78
80
82
84
86
88
90
92
94
3
2.5
2
1.5
1
0.5
0

Overlapping grids principle

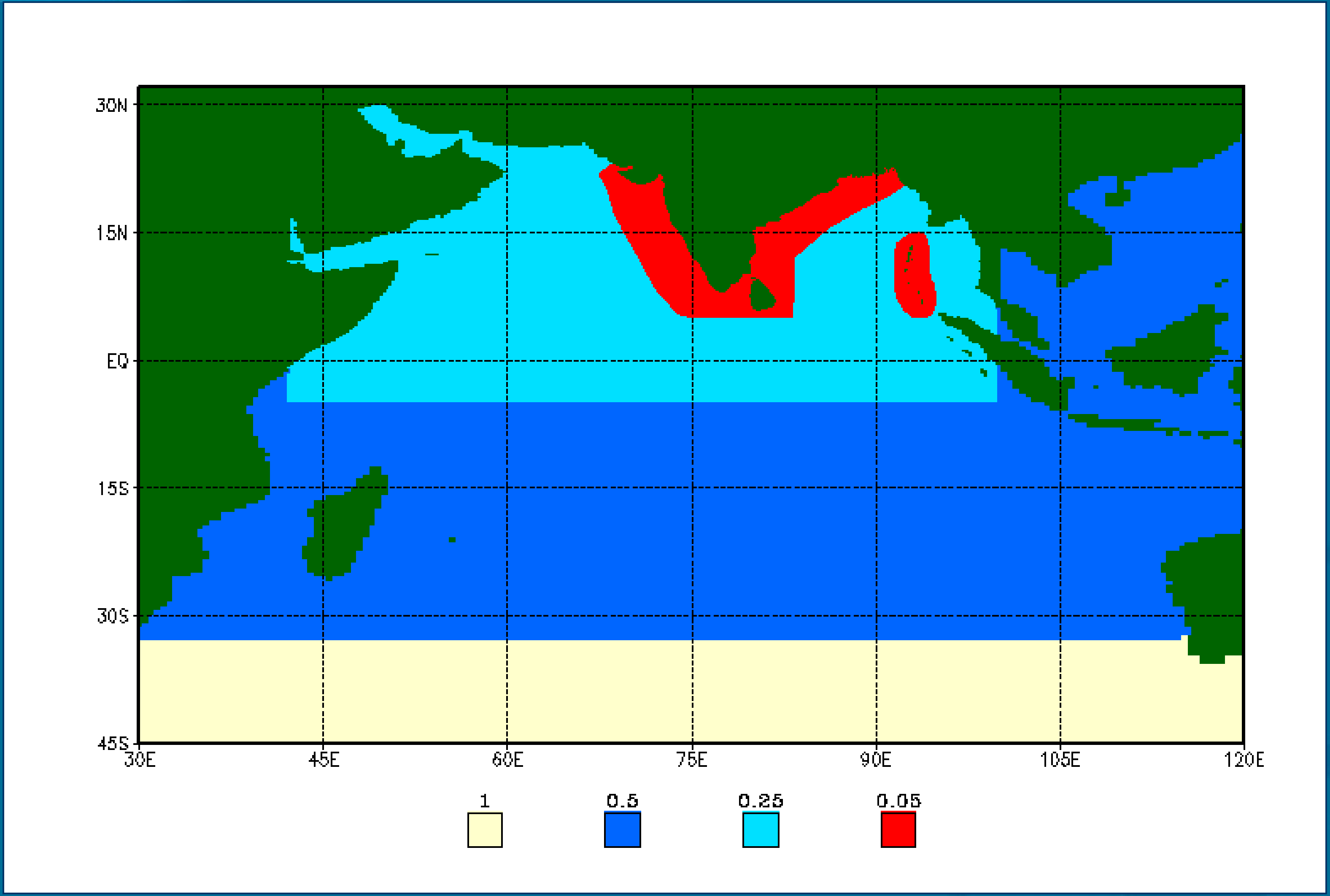
Overlaping length = stencil_width * global_time_step/CFL_time_step +
stencil_width*global_time_step/CFL_time_step+1

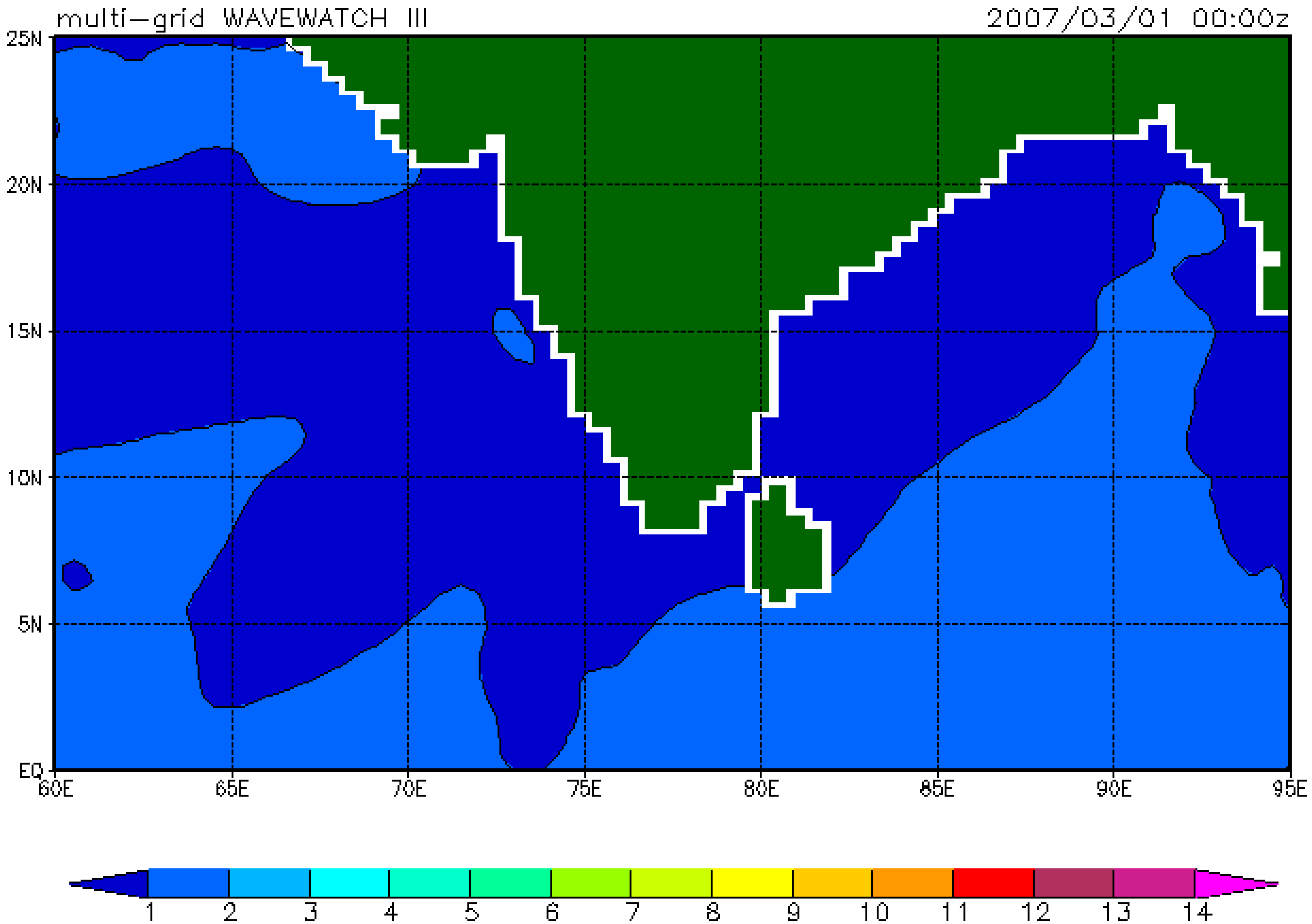


Grids for Indian Ocean forecasting System

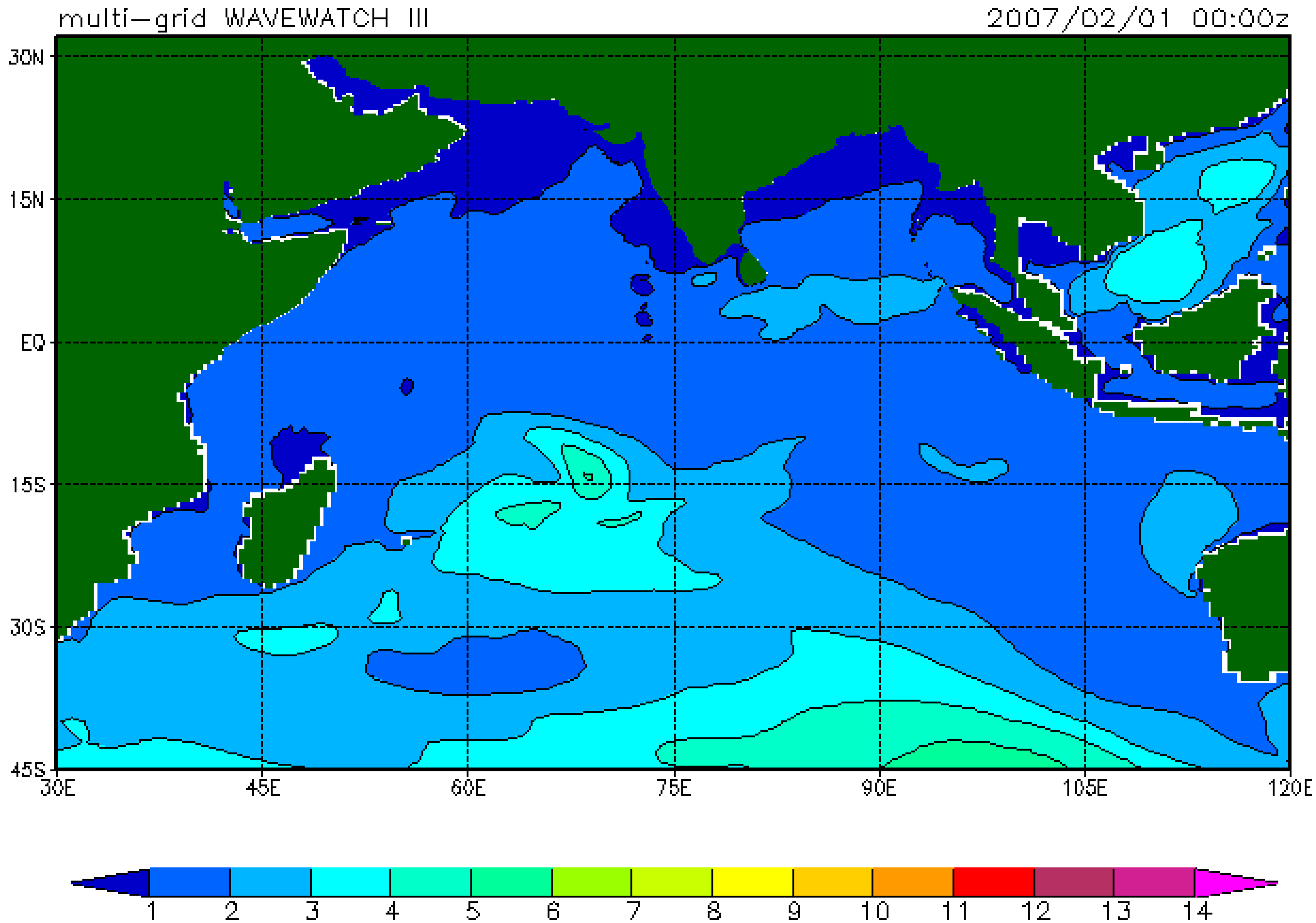
- Aim was to develop grids that account for processes from the Indian ocean but at the same time provide a high resolution coastline for the Indian subcontinent
- Developed a grid system that included
  - 1 deg Southern Hemisphere grid
  - 0.5 deg Indian Ocean grid
  - 0.25 deg Arabian Sea grid
  - 0.25 deg Bay of Bengal grid
  - 0.05 deg West Coast grid
  - 0.05 deg East Coast grid
- Month long hindcast simulation took 1976 sec (70 sec / day) CPU time on an IBM P6 super computer using 288 CPUs

Grid resolutions

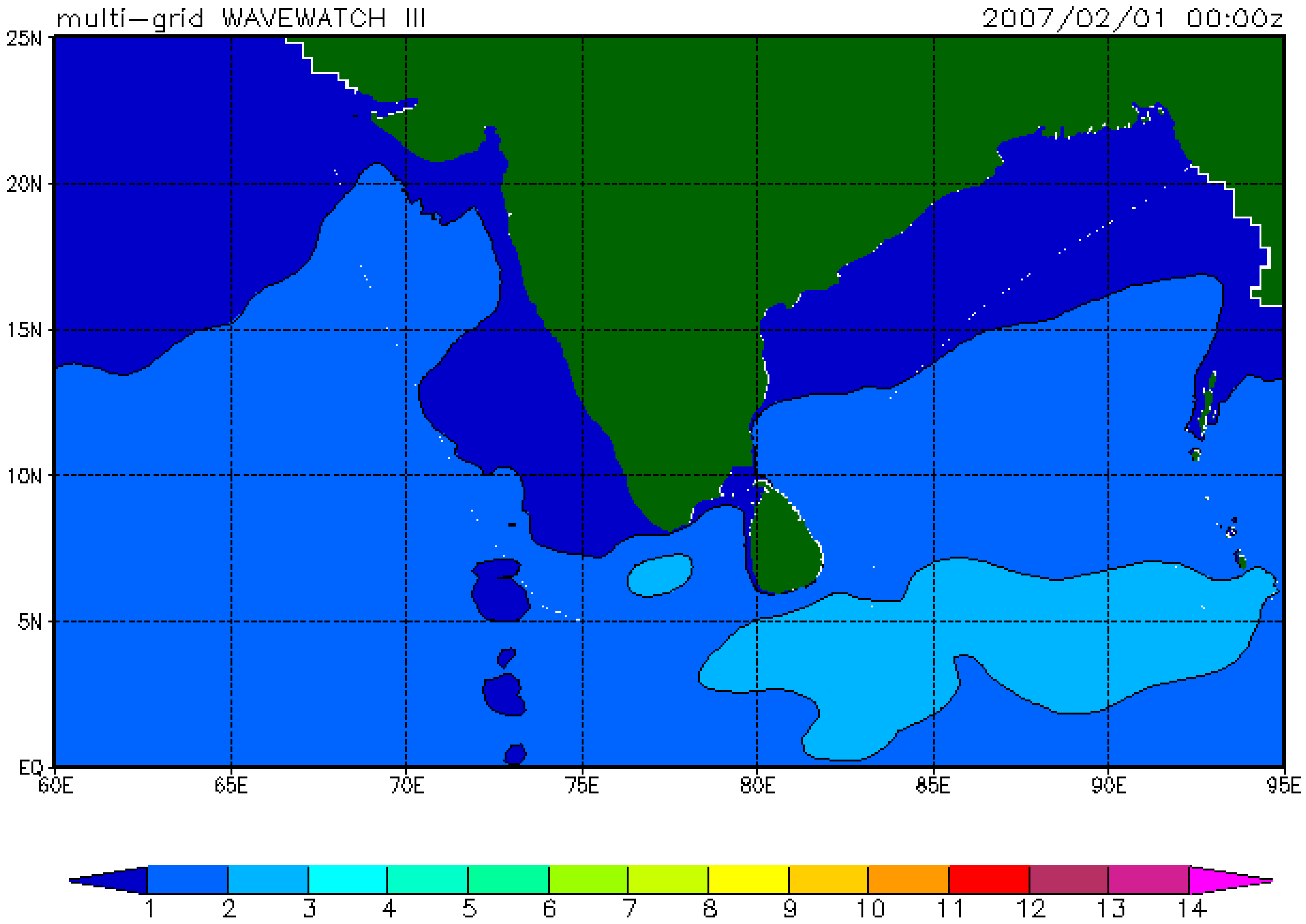



multi-grid WAVEWATCH III
2007/03/01 00:00z
25N
20N
15N
10N
5N
EQ
60E
65E
70E
75E
80E
85E
90E
95E
1 2 3 4 5 6 7 8 9 10 11 12 13 14

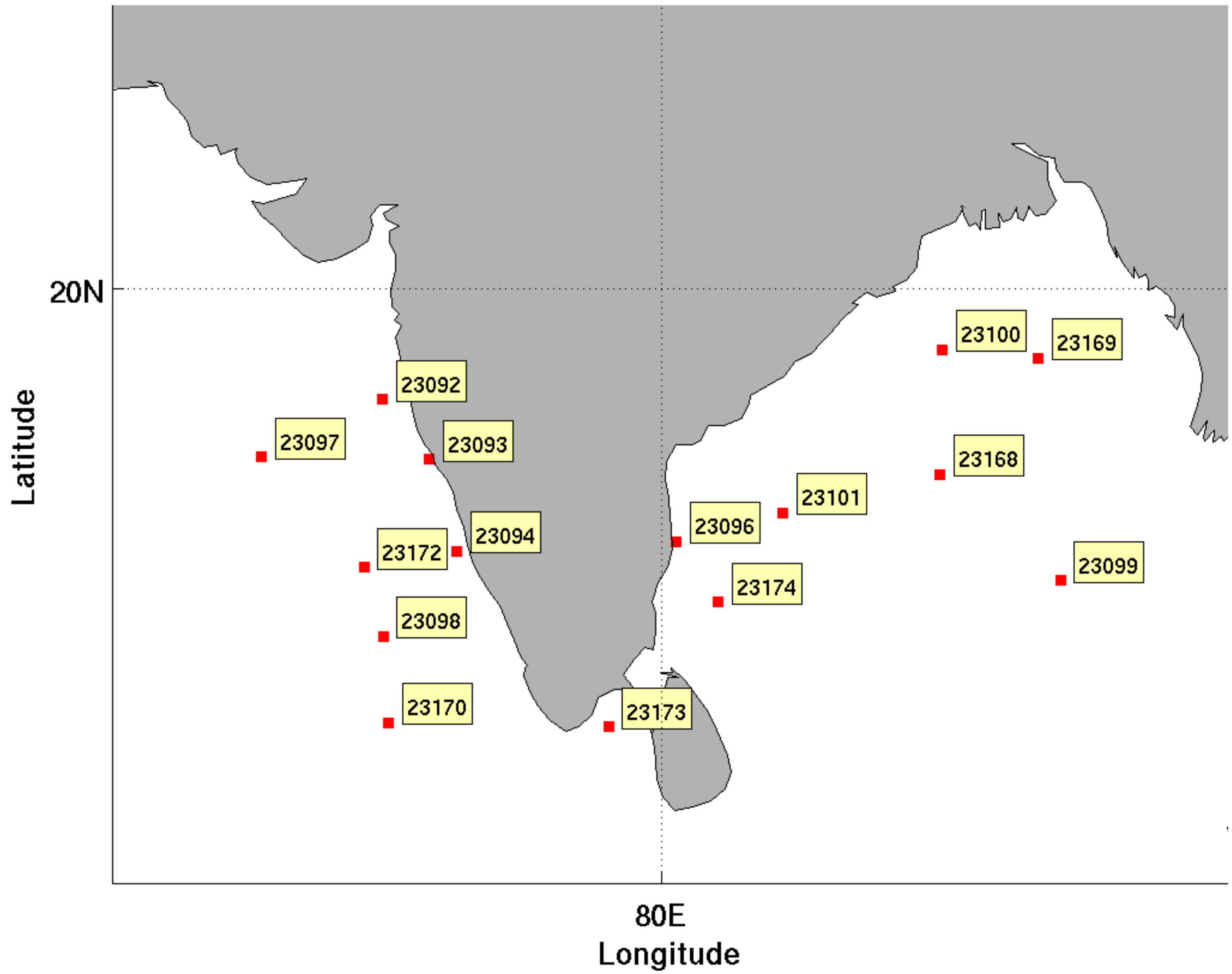
Hindcast simulation



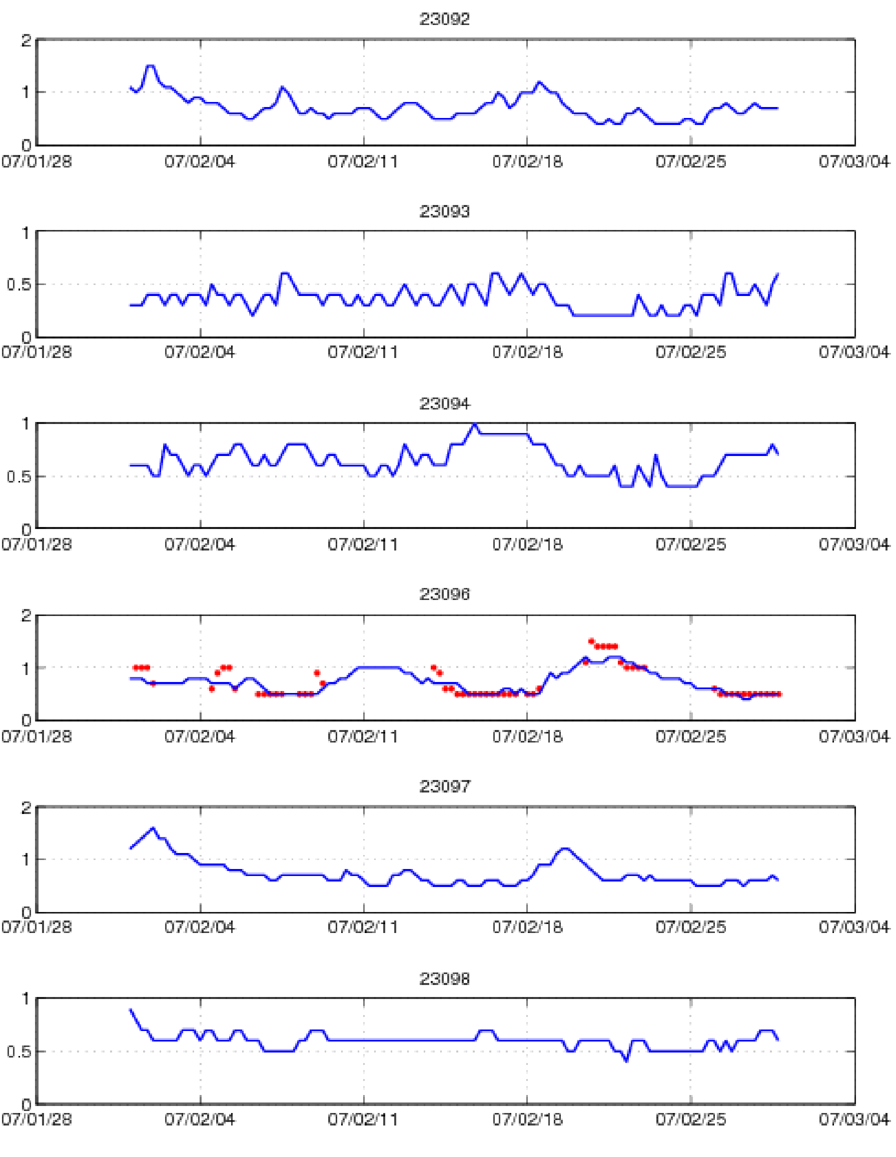
Hindcast simulation (zoom view)

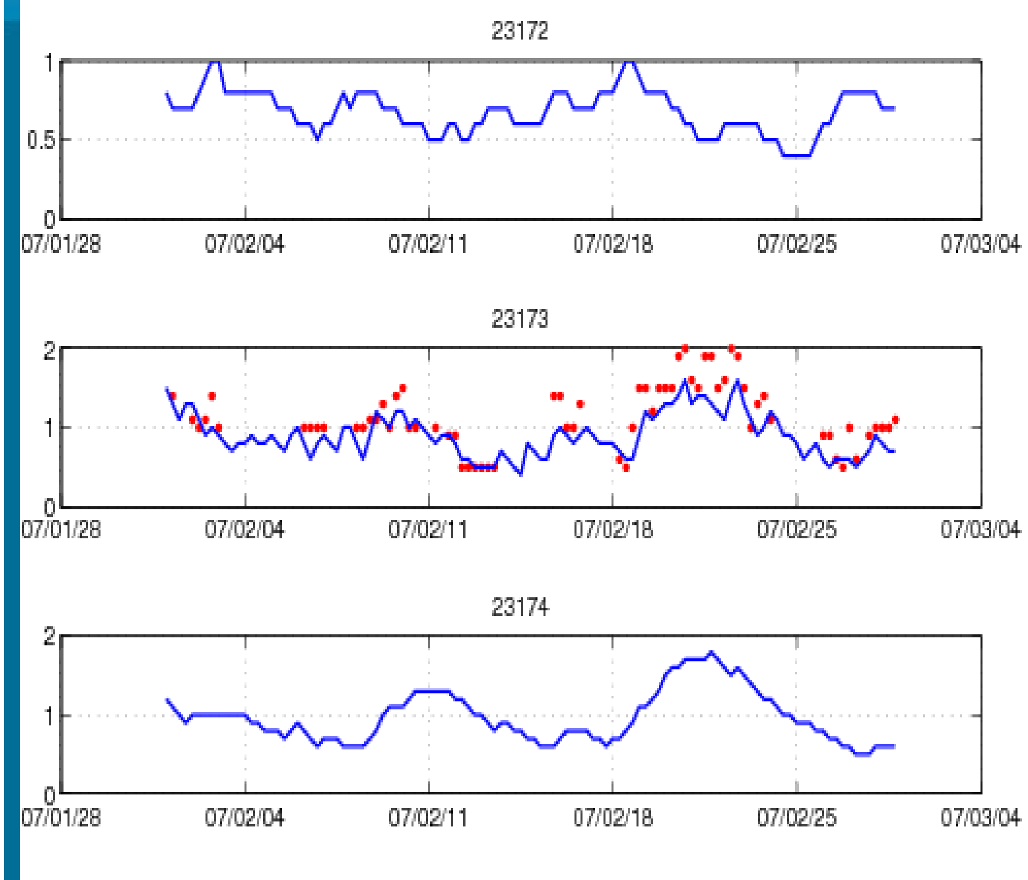


Comparisons with data



Comparisons with data





The end

End of Lecture wwws 1.2