Canadian Technical Report of
Hydrography and Ocean Sciences 222

2002

The Northwest Atlantic Tidal Current Database

by

A. Drozdowski, C.G. Hannah and J.W. Loder

Ocean Sciences Division
Maritimes Region
Fisheries and Oceans Canada

Bedford Institute of Oceanography
P.O. Box 1006
Dartmouth, N.S.
Canada B2Y 4A2

Cat. No. Fs 97-18/222E ISSN 0711-6764

Correct citation for this publication:

Drozdowski, A., C.G. Hannah and J.W. Loder, 2002. The Northwest Atlantic Tidal Current Database. Can. Tech. Rep. Hydrogr. Ocean Sci. 222: v+35 pp.

Table of Contents

List of Figures

List of Tables

Abstract

Drozdowski, A., C.G. Hannah and J.W. Loder, 2002. The Northwest Atlantic Tidal Current Database. Can. Tech. Rep. Hydrogr. Ocean Sci. 222: v+35 pp.

The Northwest Atlantic Tidal Current Database contains 5 primary tidal current constituents, M2, N2, S2, K1, O1 for the region from the Labrador Shelf to the northern New England Shelf. Many convenient search methods are included: lat-long boundary, geographic area, site name string and origin string search. This report describes the format of the data base, the data sources, and the programs that are provided for data extraction and data assimilation. The database structure is flexible and can accommodate more constituents and additional auxiliary information.

Résumé

Drozdowski, A., C.G. Hannah and J.W. Loder, 2002. The Northwest Atlantic Tidal Current Database. Can. Tech. Rep. Hydrogr. Ocean Sci. 222: v+35 pp.

La Base de Données de Courants de Marée du Nord-Ouest de l'Atlantique contient 5 composantes primaires de courants de marée, M2, N2, S2, K1 et O1 pour la région du plateau du Labrador jusqu'au plateau nord de la Nouvelle Angleterre. Beaucoup de méthodes pratiques de recherche sont incluses: recherche par limites lat-long, par région géographique, ou par chaîne de caractères de nom et de site. Ce rapport décrit le format de la base de données, la source des données, et les programmes qui sont fournis pour l'extraction et l'assimilation de données. La structure de base de données est flexible et peut être adaptée à contenir plus de composantes et des informations auxiliaires additionnelles.

1 Introduction

A comprehensive tidal current data base for the Northwest Atlantic, the NorthWest Atlantic tidal current database (NWAtcdbv1.0), has been developed. Version 1.0 covers the region from the northern New England Shelf to the Labrador Shelf, excluding the Gulf of St. Lawrence, and includes the 5 major tidal current constituents (M2, N2, S2, K1, O1) as well as the associated metadata (e.g. location, date, depth, length of record, source document). The data were extracted from standard sources such as Moody et al. (1984), from various BIO technical reports, and from recent unpublished tidal analyses. The purpose of this report is to explain the structure and usage of the database. As such, it describes the data sources, the structure of the database, and the various methods for extracting data from the database.

Section 2 describes the three components of the database: the data, the metadata and the programs. This includes a description of the data records and the metadata tags. Section 3 contains examples of the use of the database program which illustrate the ways in which the database can be queried and the procedure for adding new data. Section 4 discusses several issues involved in the creation of the database and some of the decisions that were made. Appendix A contains detailed maps to aid the user in finding desired sites.

The program for database queries was written in the *Perl 5.6.1* scripting language for a unix/linux platform. The code should be portable to the MS Windows environment provided that Perl is installed. However, at this point NWAtcdbv1.0 has not been tested on Windows. This database has an option to automatically produce location maps. In order to make use of this option, the user must have GMT (see Wessel and Smith (2001)) and GhostView installed.

2 Database

NWAtcdbv1.0 consists of 3 primary components:

1. Tidal constituents: data
2. Additional information required to give meaning to the data (e.g. latitude, longitude, water depth, date): metadata
3. Programs for accessing and manipulating the data and metadata

2.1 Data and Metadata

Tidal current velocity data is stored as east (U) and north (V) components of amplitude (cm/s) and phase lag (Degrees Greenwich). Elliptical components were not included in

Table 1: Data tag definitions

Full Name	Shortcut	Description
site_name	site	Site name string (e.g. CASP1)
file_name	file	BIO data bank file name
start_date	date	Tidal analysis start date (dd/mm/yy or mm/yy or yyyy)
days_duration	days	Length of analysis (days)
longitude	long	Longitude (decimal degrees East)
latitude	lat	Latitude (decimal degrees North)
mooring_station	mooring	Original site station number
water_depth	water	Water depth (meters)
instrument_depth	instr	Instrument depth (meters)
sampling_rate		Sampling rate (sec)
area		General geographic location description. Should contain no spaces (e.g. browns_bank_scotian_shelf)
origin		Information pointing to the data source (e.g. moody_atlas)
M2		Lunar Semi-Diurnal Component
N2		Large Lunar Elliptic Component
S2		Solar Semi-Diurnal Component
K1		Luni-Solar Diurnal Component
O1		Lunar Diurnal Component

the database. These can be generated when required (e.g. Moody et al. (1984)). Xu (2000) has written a MATLAB program to perform the task.

Metadata is a term used to describe the addition information that gives meaning to the tidal currents. Examples of metadata include site name, latitude, longitude and water depth.

The term "tag" refers to a particular name of data or metadata recognized by the computer. A tag is usually synonymous with what it represents. For example, the tag for latitude is *latitude*, and the tag for site name is *site_name*. The number of tags for each site varies depending on the availability of information. Additional tags can be added as required. For a complete list of tags and their definitions refer to Table 1. In the text to follow, tags will be distinguished with *italics*.

The term "record" refers to all the data and metadata associated with a specific geographic location (site). Table 2 shows an example of a record. All lines are of the form *tag_name: value* where *value* and *tag_name* must not include spaces with the exception of the data itself in which case four components separated by spaces are expected. The

Table 2: Example of NWAtcdbv1.0 Record Format

site_name: BED64
origin: moody_atlas
area: bay_of_fundy
longitude: -65.233
latitude: 45.217
days_duration: 29
instrument_depth: 10
water_depth: 50
M2: 101.2 16 61.3 27
N2: 20.3 351 12.3 2
S2: 15.6 63 9.5 73
K1: 2.1 122 1.5 139
O1: 1.6 94 1.1 111
&
instrument_depth: 25
M2: 85.5 25 51.4 25
N2: 17.1 360 10.3 360
S2: 13.1 72 7.9 72
K1: 2.1 125 1.2 128
O1: 1.5 97 0.9 101

components are entered in the form: (U_{amp} U_{pha} V_{amp} V_{pha}), where U_{amp} and V_{amp} are the north and east velocity amplitudes (cm/s), while U_{pha} and V_{pha} are the corresponding phase lags (degrees Greenwich). The tag *site_name* starts the record and is followed by all the data and metadata associated with that site. *site_name* is the most specific geographical location specifier aside from lat-lon coordinates. It is a useful way of describing an area of roughly a few square kilometers that contains multiple measurements which one wishes to group together for analysis. A record may contain numerous data entries with different date, depth, and lat-lon all pertaining to the same site. These data-metadata groups are separated by the ampersand (&). Only tags that vary from the above group need to be included as can be seen in the example above. The second data-metadata group includes just the *instrument_depth 25* line and assumes all other tags to be the same. This format avoids needless repetition. Furthermore, the order in which tags appear within a group is not important, however for clarity data usually follows metadata. As a minimum, a record must contain the latitude, longitude, site_name and at least one data item. The record is terminated by *site_name* starting a new record or by the end of file.

2.2 Programs

NWAtcdbv1.0 contains the main database file and interpreter, various utility programs and a complete set of source files. Table 3 gives detailed information about all included files. The database interpreter, NWAtcdbv1.0.pl and other utility programs were written in the Perl script language and require it to be installed on the user's computer.

The format for using NWAtcdbv1.0 at command line is:

`NWAtcdbv1.0.pl options < NWAtcdbv1.0.db > output.txt`

where `output.txt` is an arbitrary output file. Command line `options` are discussed in Table 4 (also see examples in section 3).

Two utility programs are used to create NWAtcdbv1.0.db from source files. The first, convert2db, takes files in line format (having the same format on each line) and turns it into a NWAtcdbv1.0 file. It was designed to handle any of the many different line formats encountered while compiling the database. The command line

`convert2db -f line_format -d db_format -a string < input > output`

converts source file `input` with columns in `line_format` into a database file with `db_format`. `input` must be header free with the same number of space separated entries on each line. `line_format` and `db_format` are comma separated lists containing the tag names corresponding to the content and order of the data and metadata in the input and output files. For example, `input` may have the columns: site name, longitude, latitude, water depth, height above bottom, instrument depth and the four M2 velocity components (U_{amp} U_{pha} V_{amp} V_{pha}). In such a case, `line_format` would be: `site_name,longitude,latitude,water_depth,hab,instrument_depth,M2`. The `-a` switch allows additional information not found in `input` to be attached to each line. Suppose, for example, that all the data in `input` comes from the Bay of Fundy. It can be included as an area tag by setting `string` to `bay_of_fundy` and including the `area` tag name at the end of the `line_format` list. The `db_format` list specifies the sequence of tags to be written to `output`. For the above example, `site_name,area,longitude,latitude, water_depth,instrument_depth,M2` would place the *area* tag right after *site_name* and exclude the redundant height above bottom *hab* tag. Table 5 contains a summary of convert2db options.

The second program, append2db, merges small databases obtained by convert2db taking care to group data from same sites together. The command

`append2db olddb < inputdb > newdb`

merges a database file `inputdb` to a pre-existing database `olddb` creating the new version `newdb`. Successive executions of this script create the entire database. The user of NWAtcdbv1.0 need not use convert2db or append2db directly. Instead, if changes have been made to the source files, the database can be recompiled by executing the build_database script.

Table 3: List of files.

Main Database Files	
NWAtcdbv1.0.pl	Perl Script used to access and search the database
NWAtcdbv1.0.db	Database
build_database	Script to compile source files
Utility Files	
convert2db	Converts line format data found into source database file
append2db	Merges smaller database files
Siteanalysis	Program to determine the scatter in location and depth of sites
Source Files	
AO1987	Petrie et al. (1987)
AO1987hibernia	Petrie et al. (1987)
CDRno62	Wright et al. (1988)
CDRno71	Ross et al. (1988)
CDRno91	Lively and Petrie (1990)
CDRno94	Loder and Pettipas (1991)
CDRno96	Wright et al. (1991)
CDRno102	Petrie (1991)
CTRno40	Lively (1984a)
CTRno46	Lively (1984b)
CTRno66	Lively (1985)
CTRno100	Lively (1988)
CTRno113	Lively (1989)
CTRno157	Narayanan (1994)
moody_atlas.txt	Moody et al. (1984)
nfld_shelf.txt	Shawn Oakey(pers. comm. 2002)
scotian_shelf_tidal.txt	Shawn Oakey(pers. comm. 2002)
unknown	Records of untraced origin

Table 4: List of command line options for NWAtcdbv1.0.pl.

Option	Explanation
-f *list*	Comma separated *list* of tags desired in the output file (e.g. -f *site_name,latitude,longitude*). Each item is formatted to the length of longest string in that column and left aligned. To choose the length for a column simply type the number of characters after a colon following the tag name (e.g. -f *site_name,latitude:7,longitude:7*). Data will be truncated to fit in specified column width.
-s *string*	Gives data for sites that match *string* (e.g. -s *C1*). Wildcards may be used.
-o *string*	Gives data for sites with *origin* matching *string* (e.g. -o *moody*). Wildcards may be used.
-a *string*	Gives data for sites with *area* matching *string* (e.g. -a *labrador*). Wildcards may be used.
-c *list*	Gives data for sites matching all tidal components in *list* (e.g. -c *M2,N2*). Wildcards may be used.
-v	Produces a simple plot of outputed site locations. Requires GMT (see Wessel and Smith (2001)) and GhostView
-l *x1,y1,x2,y2*	Selects sites falling within the rectangle defined by bottom left *(x1,y1)* and top right *(x2,y2)*. The coordinates must be supplied in decimal degrees east and north.

build_database is a Perl script which uses convert2db and append2db to assemble the source files (see table 3). It requires no arguments or options. Execution of this script produces a new database called .db which should be appropriately renamed.

Table 5: Summary of convert2db options.

Option	Explanation
-f *list*	Comma separated *list* of tag names describing each column of the input file. The names are chosen by the user but once chosen must be kept contant as the database referes to each data type by that name. (e.g. site_name,start_date,M2,S2)
-d *list*	Comma separated *list* of tags desired in the ouput database. tags names must match those supplied in the -f option, however, tags can be omitted and shuffled around. The purpose of this is to exclude unwanted data from the database and control the order in which data tags appear. (e.g. site_name,S2,M2)
-a *string*	Additional tag to adjoin to input. The purpose of this option is to allow the user to input an additional tag common to the entire input file. Note: matching tag name must appear at the end of the -f *list*. (e.g. bay_of_fundy)

3 Examples

This section contains several examples of using NWAtcdbv1.0, convert2db and append2db. They serve to both demonstrate the usage of the scripts and to offer test cases which may be used for establishing validity of database. All of the files mentioned here are included with the distribution of the database (see Appendix B).

3.1 NWAtcdbv1.0.pl

a) To get a brief summary of available options and tag name short cuts:

```
NWAtcdbv1.0.pl
```

b) To see a list of available areas in the database:

```
NWAtcdbv1.0.pl -f area  < NWAtcdbv1.0.db
```

which produces the output:

area
bay_of_fundy
georges_bank
great_south_channel
gulf_of_main

labrador
nantucket_shoals
new_england_shelf
newfoundland
newfoundland_avalon_channel
newfoundland_southeast_shoal
northeast_channel
scotian_shelf

c) To get the M2 constituent for all available sites in the Bay of Fundy:

```
NWAtcdbv1.0.pl -a bay_of_fundy -f site,long,lat,water:4,instr:4,
M2:8 < NWAtcdbv1.0.db
```

which produces the output:

site_	longitu	latitu	wate	inst	M2ampu	M2phau	M2ampv	M2phav
BED60	-66.400	45.000	84	13	64.6	22	32.8	41
BED61	-66.200	44.817	107	13	77.0	18	26.1	35
BED61	-66.200	44.817	107	50	80.0	14	34.3	23
BED62	-66.033	44.650	90	13	101.2	14	42.7	33
BED63	-65.333	45.317	50	25	78.1	20	43.3	20
BED64	-65.233	45.217	50	10	101.2	16	61.3	27
BED64	-65.233	45.217	50	25	85.5	25	51.4	25
BED65	-65.133	45.133	62	25	82.0	20	64.3	22
BED66	-65.117	45.417	38	25	62.8	17	36.4	22

d) To get S2 and K1 for all the data from Narayanan (1994):

```
NWAtcdbv1.0.pl -o report_no.157 -f site,S2:6,K1:6
< NWAtcdbv1.0.db
```

which produces the output:

site	S2ampu	S2phau	S2ampv	S2phav	K1ampu	K1phau	K1ampv	K1phav
BB1	1.8	86.3	0.8	349.7	1.56	301.9	1.5	204.1
BB1	2.1	91.8	0.84	358.7	1.0	276.1	1.1	182.8
BB2	1.4	97.2	0.67	16.2	1.02	289.54	1.11	163.37
BB2	2.0	86.1	0.85	357.4	0.8	297.2	0.88	189.15
BB4	0.6	70.1	0.3	240.7	0.55	280.8	0.37	182.3
BB4	0.65	89.8	0.1	195.4	0.5	276.8	0.16	166.3
BB4	0.76	86.3	0.09	300.8	0.57	271.7	0.24	187.5
FI3	2.0	85.2	1.03	338.8	1.45	259	1.7	161.4
FI3	2.02	92	1.04	11.5	1.2	240.2	1.7	146.6

```
NWB1 1.53   90.5   0.6    352.5  0.09   301.1  0.2    111.1
NWB1 2.35   92.3   1.33   15.5   0.15   109.3  0.5    105.7
```

e) To get all the available sites along with their originating publications for a rectangular area in the vicinity of the Avalon Peninsula specified by long-lat bottom left and top right corners (-54,47) and (-51,48):

```
NWAtcdbv1.0.pl -l -54 47 -51 48 -f site,origin < NWAtcdbv1.0.db
```

which produces the output:

```
site_ origin
AVAL3 atmosphere_ocean_1987_B._Petrie_et._al.
AVAL3 canadian_data_report_no.102_B.Petrie_1991
AVAL4 atmosphere_ocean_1987_B._Petrie_et._al.
AVAL4 canadian_data_report_no.102_B.Petrie_1991
AVAL5 atmosphere_ocean_1987_B._Petrie_et._al.
AVAL5 canadian_data_report_no.102_B.Petrie_1991
BB2   canadian_technical_report_no.157_S.Narayanan_1994
NFLD1 canadian_data_report_no.62_d.g.wright_et.al._1985-6
```

3.2 convert2db

a) The program convert2db converts the line format file AO1987 below

```
Linnet  NA newfoundland -50.424 48.208 160 NA   103 8.5 68 5.7 356
AVAL3 393 newfoundland_avalon_channel -52.582 47.501 173 80   123
0.7 60 8.5 60
AVAL4 394 newfoundland_avalon_channel -52.315 47.432 160 63.3 123
3.0 78 7.5 58
```

with the following script named convertAO1987_2db:

```
#!/usr/bin/perl
$ori = "atmosphere_ocean_1987_B._Petrie_et._al.";
$f = "AO1987";
$format_array ="site_name,mooring_station,area,longitude,latitude,
water_depth,instrument_depth,days_duration,M2,origin";
$db_format = "site_name,mooring_station,origin,area,longitude,latitude,
water_depth,instrument_depth,days_duration,M2";
system( "./convert2db -f $format_array -d $db_format -a $ori < $f
> AO1987.db" );
```

The output is a NWAtcdbv1.0 format file AO1987.db:

```
site_name: Linnet
mooring_station: NA
origin: atmosphere_ocean_1987_B._Petrie_et._al.
area: newfoundland
longitude: -50.424
latitude: 48.208
water_depth: 160
instrument_depth: NA
days_duration: 103
M2: 8.5  68  5.7  356

site_name: AVAL3
mooring_station: 393
origin: atmosphere_ocean_1987_B._Petrie_et._al.
area: newfoundland_avalon_channel
longitude: -52.582
latitude: 47.501
water_depth: 173
instrument_depth: 80
days_duration: 123
M2: 0.7  60  8.5  60

site_name: AVAL4
mooring_station: 394
origin: atmosphere_ocean_1987_B._Petrie_et._al.
area: newfoundland_avalon_channel
longitude: -52.315
latitude: 47.432
water_depth: 160
instrument_depth: 63.3
days_duration: 123
M2: 3.0  78  7.5  58
```

b) Another example of convert2db

aval3.txt:

```
AVAL3 393 newfoundland_avalon_channel 21/6/80 -52.583 47.501 175.0
130.0 124.0 0.6 38.55 8.1 58.1 0.6 359.3 1.2 34.3 0.3 111.3 1.5
132.7 0.3 117 0.85 297 0.1 160.35 1.5 277.9
```

convertaval3_2db:

```
#!/usr/bin/perl
$ori="canadian_data_report_no.102_B.Petrie_1991";
$f= "aval3.txt";
$format_array ="site_name,mooring_station,area,start_date,longitude,
latitude,
water_depth,instrument_depth,days_duration,M2,N2,S2,K1,O1,origin";
$db_format = "site_name,origin,area,longitude,latitude,water_depth,
instrument_depth,
start_date,days_duration,M2,N2,S2,K1,O1";
system( "./convert2db -f $format_array -d $db_format -a $ori < $f
> aval3.db" );
```

aval3.db:

```
site_name: AVAL3
origin: canadian_data_report_no.102_B.Petrie_1991
area: newfoundland_avalon_channel
longitude: -52.583
latitude: 47.501
water_depth: 175.0
instrument_depth: 130.0
start_date: 21/6/80
days_duration: 124.0
M2: 0.6  38.55  8.1  58.1
N2: 0.6  359.3  1.2  34.3
S2: 0.3  111.3  1.5  132.7
K1: 0.3  117  0.85  297
O1: 0.1  160.35  1.5  277.9
```

3.3 append2db

The two database files obtained from the previous examples, AO1987.db and aval.db, can now be merged with append2db:

`append2db AO1987.db < aval3.db > AO1987aval3.db`

which gives one complete database file:

```
site_name: Linnet
mooring_station: NA
```

origin: atmosphere_ocean_1987_B._Petrie_et._al.
area: newfoundland
longitude: -50.424
latitude: 48.208
water_depth: 160
instrument_depth: NA
days_duration: 103
M2: 8.5 68 5.7 356

site_name: AVAL3
mooring_station: 393
origin: atmosphere_ocean_1987_B._Petrie_et._al.
area: newfoundland_avalon_channel
longitude: -52.582
latitude: 47.501
water_depth: 173
instrument_depth: 80
days_duration: 123
M2: 0.7 60 8.5 60
&
origin: canadian_data_report_no.102_B.Petrie_1991
instrument_depth: 130.0
water_depth: 175.0
longitude: -52.583
start_date: 21/6/80
days_duration: 124.0
N2: 0.6 359.3 1.2 34.3
O1: 0.1 160.35 1.5 277.9
K1: 0.3 117 0.85 297
S2: 0.3 111.3 1.5 132.7
M2: 0.6 38.55 8.1 58.1

site_name: AVAL4
mooring_station: 394
origin: atmosphere_ocean_1987_B._Petrie_et._al.
area: newfoundland_avalon_channel
longitude: -52.315
latitude: 47.432
water_depth: 160
instrument_depth: 63.3
days_duration: 123
M2: 3.0 78 7.5 58

4 Remarks

The goal of this project was to create, during the period of Jan.-Mar. 2002, a comprehensive data bank of the tidal current constituents for the Northwest Atlantic. Emphasis was placed on creating a format which could be easily expanded in the future to include more information.

NWAtcdbv1.0 was assembled from various sources including complete and incomplete data files. Whenever possible an effort was made to find the original documents in order to verify that the data in the files was correct and complete. Wherever it was fragmented, as was the case with the data files related to Moody et al. (1984), the missing information was entered manually. Data were logically divided up into files by origin and region. This was done so that if doubt concerning the validity of a record arises in the future, it can be easily traced to its source. Also as the database grew in size and geographic extent it became useful to be able to visualize the locations of the data. A plotting routine was implemented in the Perl script NWAtcdbv1.0.pl which can be invoked with the "-v" option.

The first step in the creation of the database was developing a system for data storage and retrieval. The original database, developed by Paul Chapman and maintained by Shawn Oakey was taken as a starting point and gradually modified to fit the needs of the present project. The most time consuming part of the project was converting the various files and data reports into one format. Two Perl scripts were created for this purpose: convert2db and append2db.

Early on in the project it was decided that all data would be stored only in amplitude-phase format. The original files and reports contained amplitude-phase and/or elliptical coefficients. This meant that a large part of the data had to be transformed. What made the task more difficult was dealing with two different conventions of elliptic coordinates. One, the Forman convention (Foreman (1978)), has minor axis, eccentricity, inclination and phase, the other, usually found in text books and publications, has the major axis, minor axis, orientation and phase. The units rarely posed a problem typically having north and east as the positive axis and abiding by Greenwich time. The distinction between inclination and orientation did not seem obvious at first but later it was discovered that the former measures angles counter clockwise from the east axis while the latter clockwise from the north.

NWAtcdbv1.0 was designed to be open to future development, therefore it was necessary to assign it a version number. Upgrades in the database file NWAtcdbv1.0.db will evolve separately from the interpretor script file NWAtcdbv1.0.pl with one exception: the first digit in the version number, `a` of `va.b`, must agree between the two. What this means is that increments in `a` must reflect compatability issues. Once NWAtcdbv1.0.pl has been altered to the point that the old NWAtcdbv1.0.db format is unrecognizable, `a` must be incremented in both files together. Once `a` is incremented, convert2db and append2db must also be altered to reflect the format change.

5 Conclusions

A tidal current database for the Northwest Atlantic was compiled from available BIO reports, cruises and publications. Version 1.0 includes sites ranging from the Northern New England shelf to as far north as Labrador (See Figures 1 - 7). Most sites contain the five dominant tidal constituents: M2, N2, S2, K1, O1 plus all the associated site specific information. Due to the flexibility of the format in which the data was stored, the database can easily be expended to include more data. The database can be searched by many convenient methods including lat-long boundary, geographic area, site name string and origin string search. Future work will include the enlargement of the databank and adapting NWAtcdbv1.0 for internet access.

Acknowledgements The authors would like extend warm thanks to all those who contributed to the creation of NWAtcdbv1.0, in particular Paul Chapman and Shawn Oakey for creating and maintaining the tidal current database which was our starting point. Thanks are also due to Dave Greenberg and Zhigang Xu for their support through out this project. This work was supported in part by the Canadian Hydrographic Service and the PERD Offshore Environmental Factors Program.

References

Foreman M.G.G 1978. Manual for Tidal Currents Analysis and Prediction. Pacific Marine Science Report 78-6: iii + 57 pp.

Lively, R.R. 1984a. Current meter, meteorological, and sea-level observations off Cape Sable, Nova Scotia. Can. Tech. Rep. Hydrogr. Ocean Sci. 40: v + 494 pp.

Lively, R.R. 1984b. Current meter and tide gauge observatioins for the Strait of Belle Isle. Can. Tech. Rep. Hydrogr. Ocean Sci. 46: vi + 166 pp.

Lively, R.R. 1985. Current meter, meteorological, and sea-level observations off Cape Sable, Nova Scotia, August 1980 to April 1983. Can. Tech. Rep. Hydrogr. Ocean Sci. 66: v + 526 pp.

Lively, R.R. 1988. Current meter, meteorological, and sea-level observations for the CASP experiment of the coast of Nova Scotia November 1985 to April 1986. Can. Tech. Rep. Hydrogr. Ocean Sci. No. 100: vii + 428 pp.

Lively, R.R. 1989. Current meter, meteorological, and sea-level observations for Browns Bank, Nova Scotia, April 1983 to May 1985. Can. Tech. Rep. Hydrogr. Ocean Sci. 113: 304 pp.

Lively, R.R., Petrie, B. 1990. Labrador current variability study: current meter observations October 1985 to January 1986. Can. Data Rep. Hydrogr. Ocean Sci. 91: v + 146 pp.

Loder, J.W., R.G. Pettipas 1991. Moored current and hydrographic measurements from the Georges Bank Frontal Study 1988-89. Can. Data Rep. Hydrogr. Ocean Sci. 94: iv + 139 pp.

Moody, J. A., B. Butman, R. C. Beardsley, W. S. Brown, P. Daifuku, J. D. Irish, D. A. Mayer, H. O. Mofjeld, B. Petrie, S. Ramp, P. C. Smith, and W. R. Wright, 1984: Atlas of tidal elevation and current observations on the Northeast American Continental Shelf and Slope. *U. S. Geological Survey Bulletin 1611*, 122 pp.

Narayanan, S. 1994. Current meter observatioins from Hamilton Bank and NE Newfoundland Shelf 1990 to 1993. Can. Tech. Rep. Hydrogr. Ocean Sci. 157: v + 184 pp.

Petrie, B., Lank, K., deMargerie, S. 1987. Tides on the Newfoundland Grand Banks. Atmosphere-Ocean. 25: 10-21 pp.

Petrie, B. 1991. Current meter and tidal gauge observations from Avalon Channel 1980-1981. Can. Dat. Rep. Hydrogr. Ocean Sci. 102: v + 89 pp.

Ross, C.K., Loder, J.W., Graca, M.J. 1988. Moored currents and hydrographic measurements on the southeast shoal of the Grand Bank 1986 and 1987. Can. Dat. Rep. Hydrogr. Ocean Sci. 71: 132 pp.

Wessel, P. and Smith, W.H.F. 2001. The Generic Mapping Tools (GMT) Version 3.4. http://gmt.soest.hawaii.edu/.

Wright, D.G., Lazier, J.R.N, Armstrong, W. 1988. Moored current and pressure data from the Labrador/Newfoundland Shelf June 1985 to July 1987. Can. Data Rep. Hydrogr. Ocean Sci. 62: 258 pp.

Wright, D.G., Lazier, J.R.N., Graca, M. J. 1991. Current meter and pressure data between Hamilton Bank and OWS Bravo July 1987 to August 1988. Can. Dat. Rep. Hydrogr. Ocean Sci. 96: vii + 101 pp.

Xu, Z. 2000. Ellipse Parameters Conversion and Vertical Velocity Profiles for Tidal Currents. Unpublished manuscript available from http://sea-mat.whoi.edu/tidal_ellipse-html/index.html.

A Location Maps

This appendix is a collection of maps that show the status of the NWAtcdbv1.0 in the spring of 2002. There are two types of maps: 1) overview maps that show the data coverage for large regions and 2) maps at a scale that allow us to put the site names on the map. All of the site names in the data base are on a map. The maps were generated using GMT, but we note that the -v option for NWAtcdbv1.0.pl does not plot the site names.

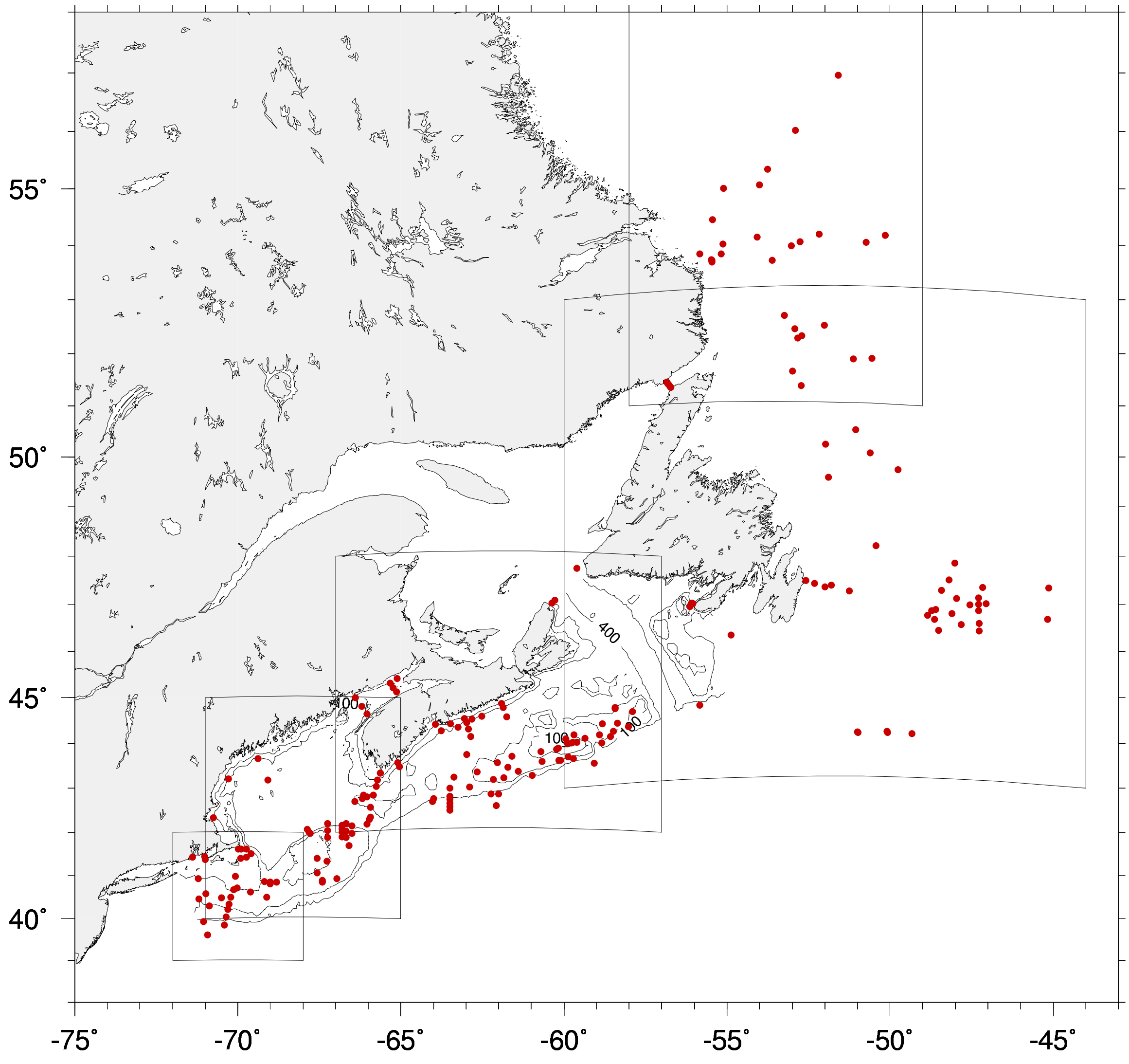

Figure 1: North West Atlantic

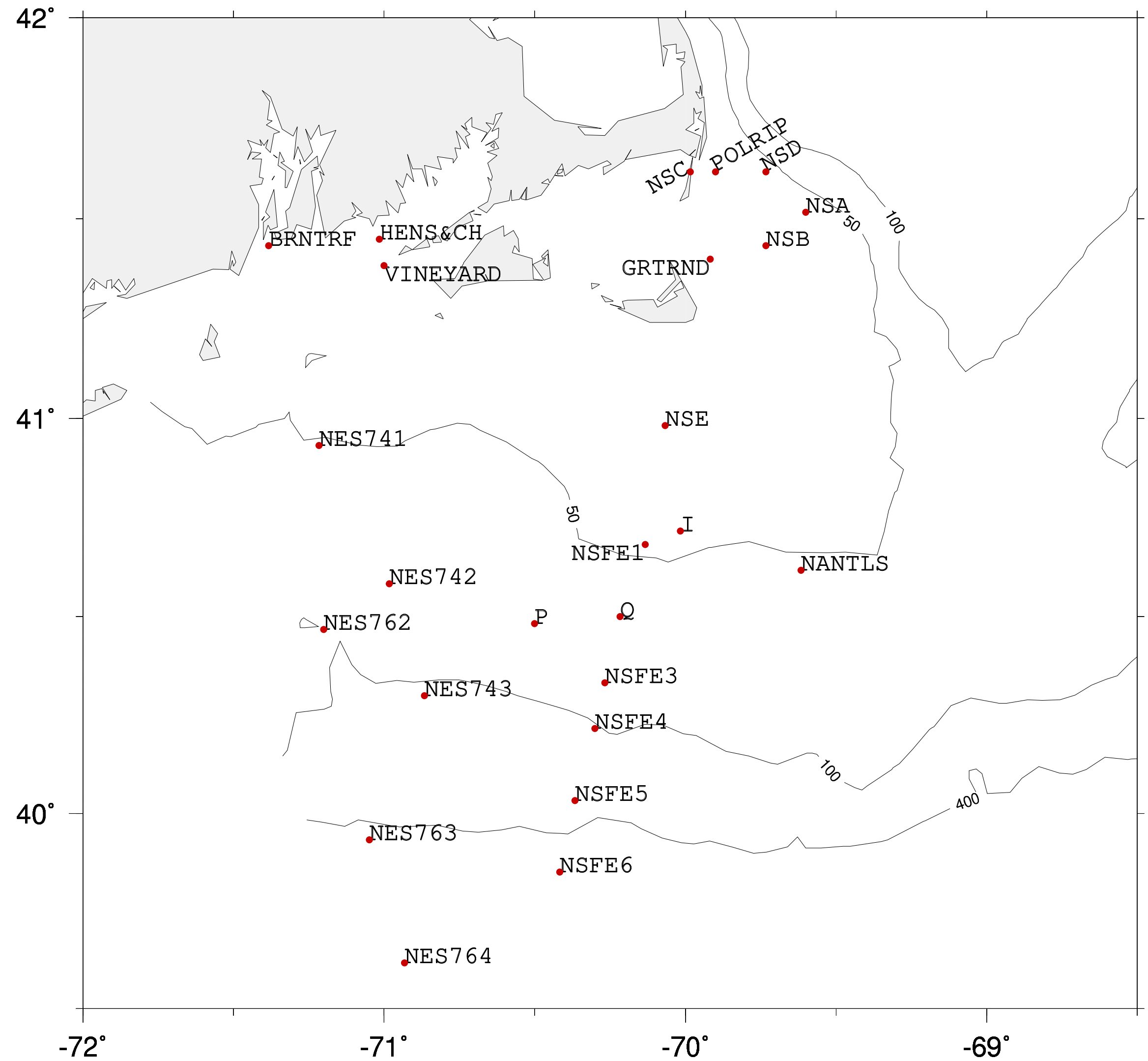

Figure 2: Northern New England Shelf

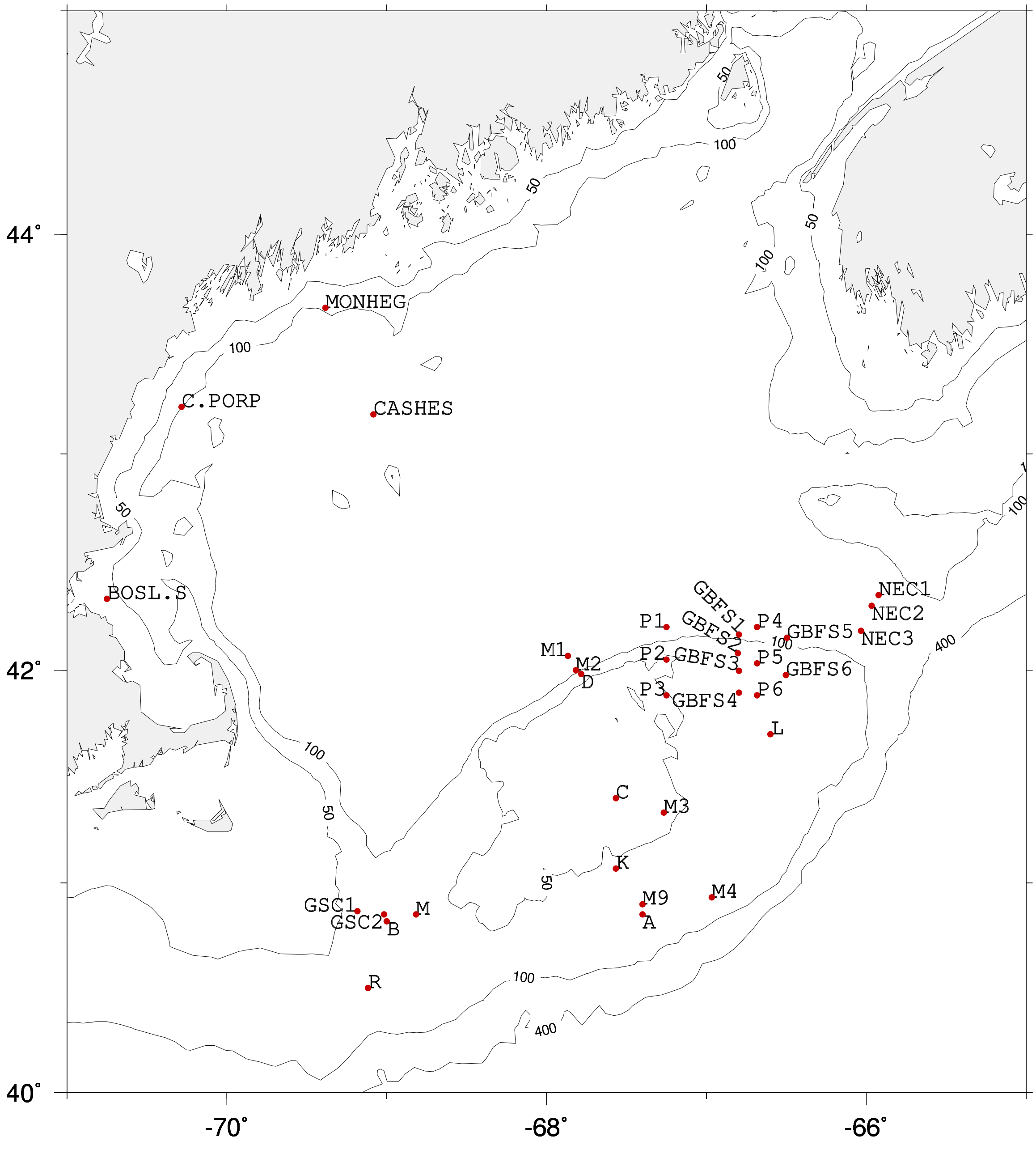

Figure 3: Gulf of Maine

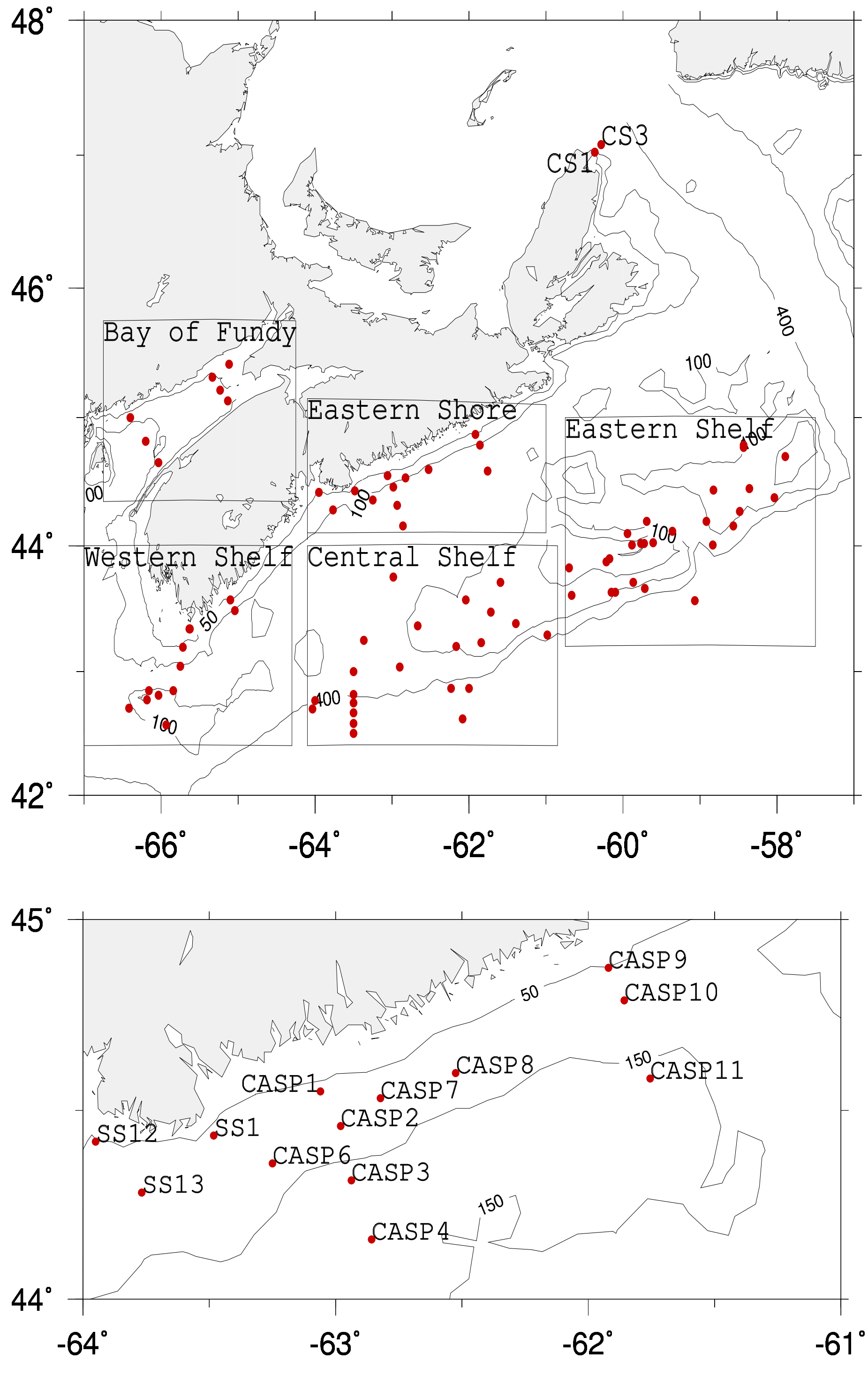

Figure 4: Scotian Shelf (top), Eastern Shore (bottom)

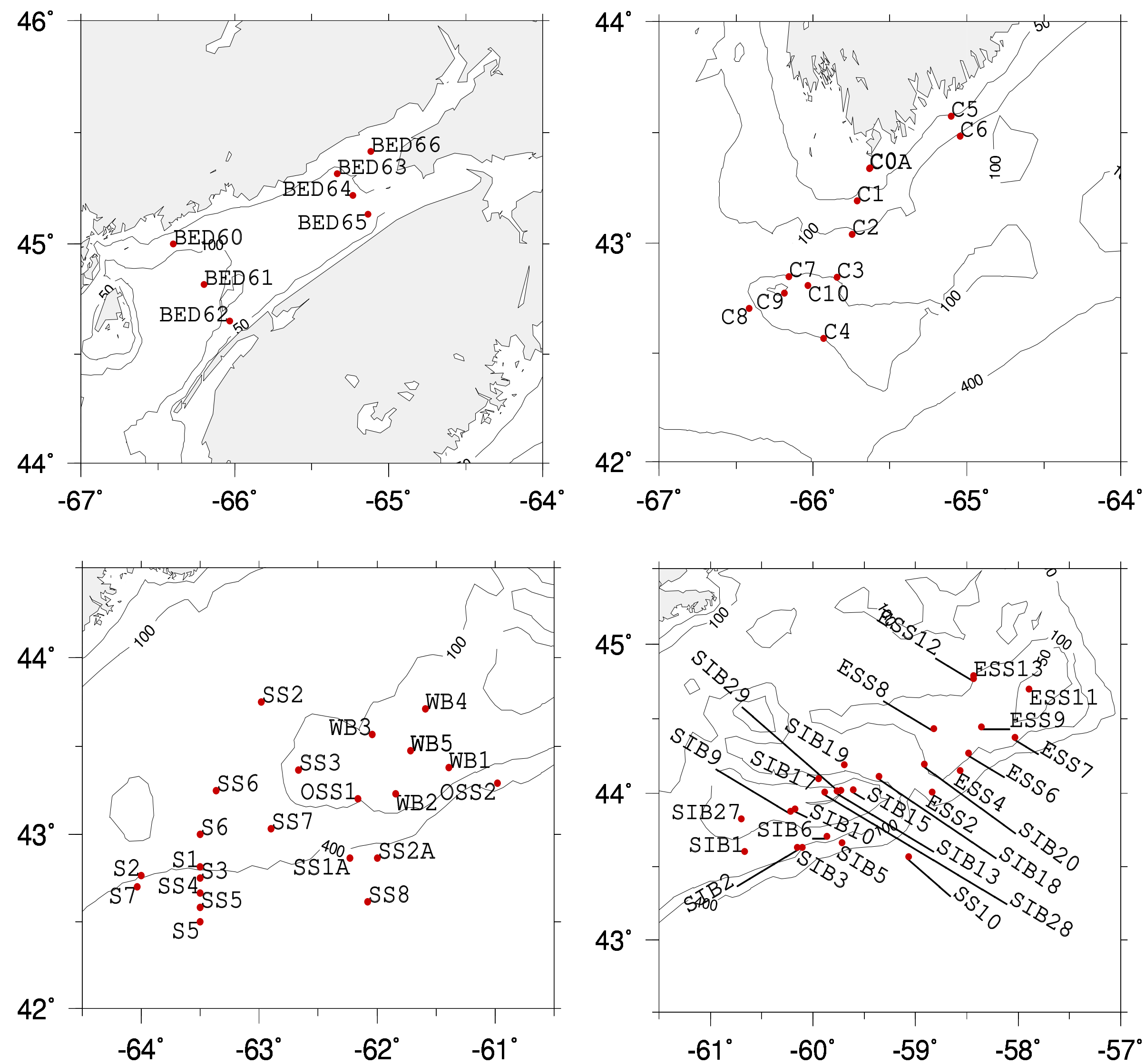

Figure 5: Bay of Fundy (top left), Western (top right), Central (bottom left)and Eastern (bottom right) Scotian Shelf

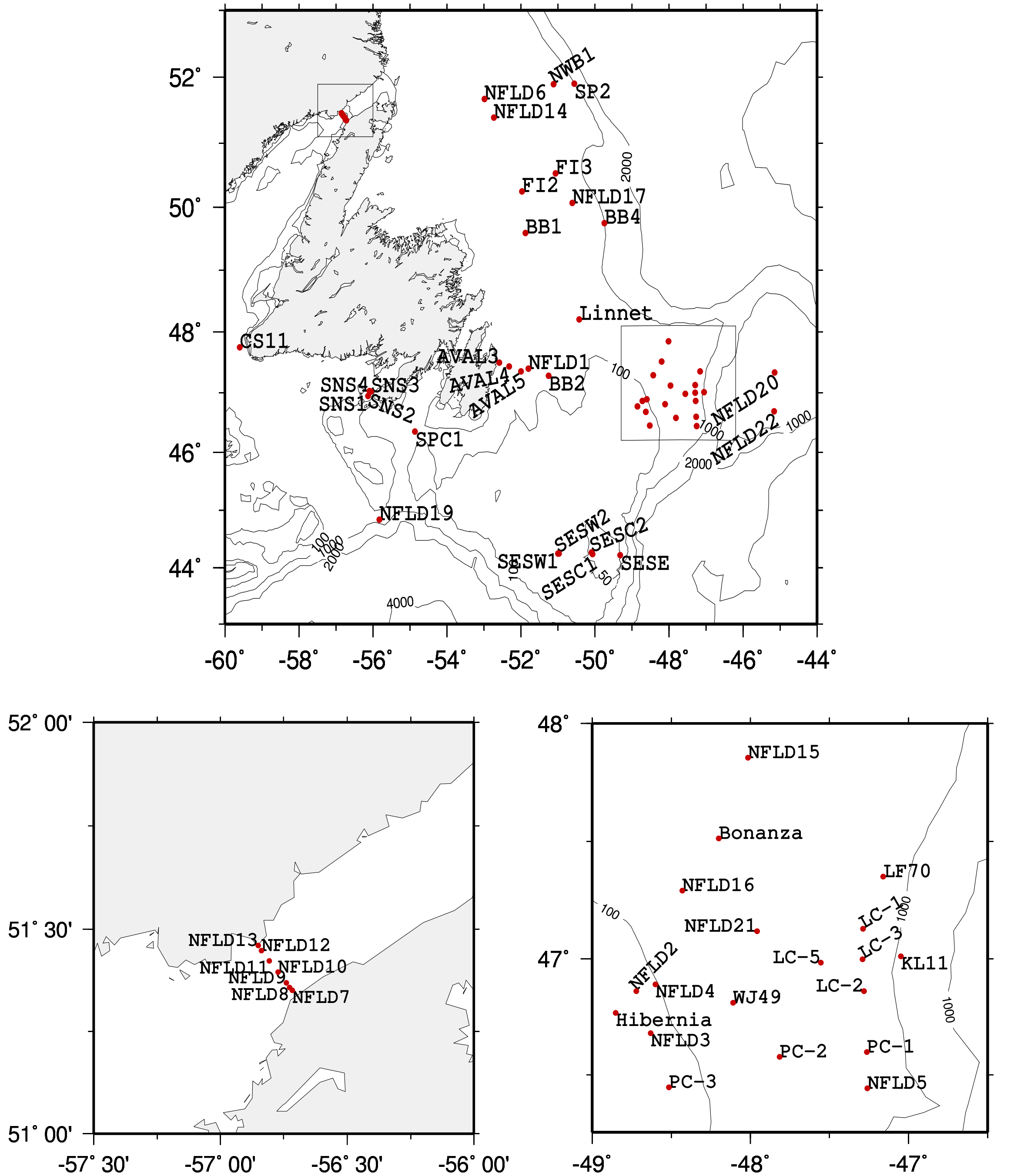

Figure 6: Newfoundland Shelf(top), Strait of Belle Isle (bottom left), Grand Banks (bottom right)

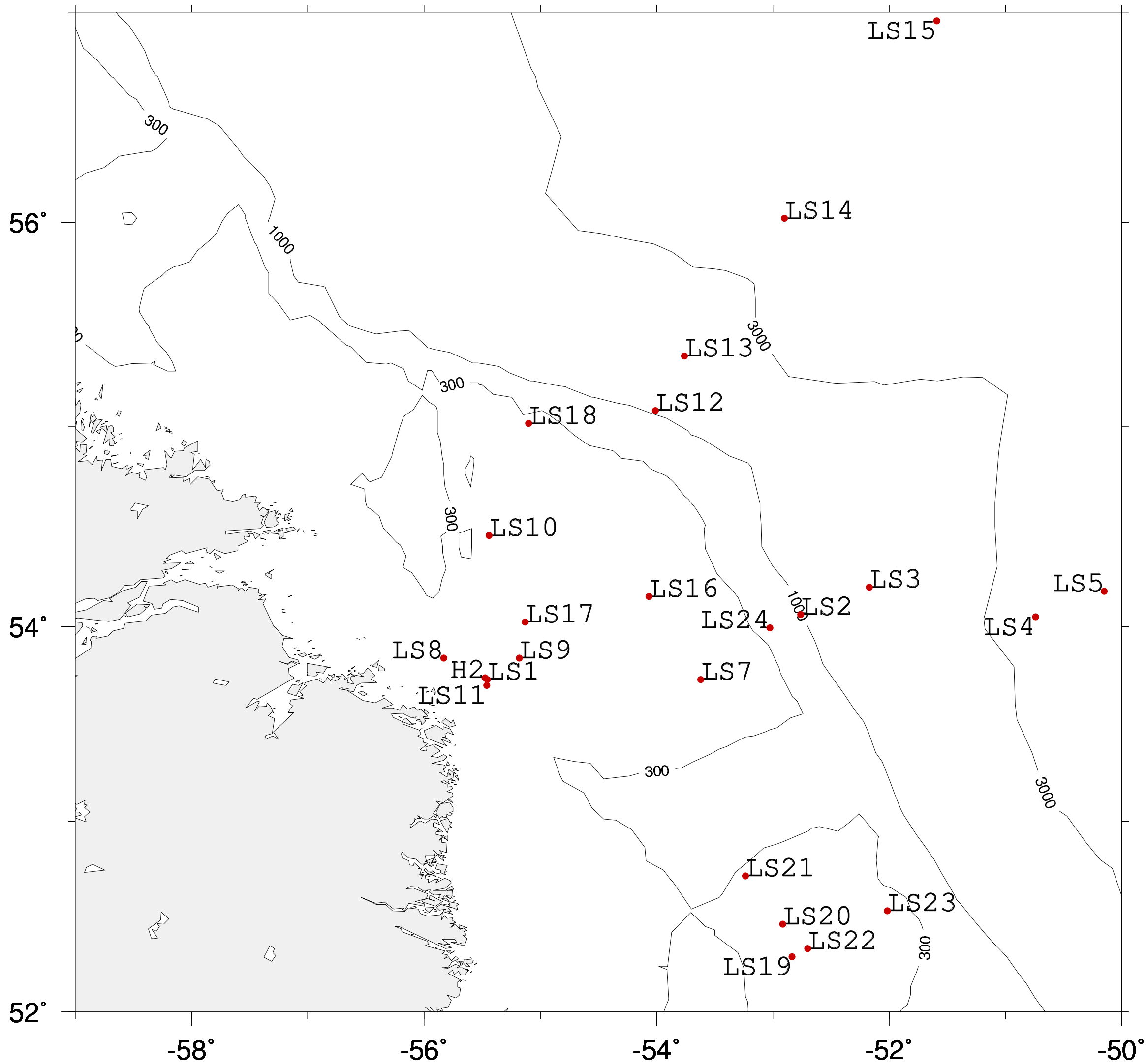

Figure 7: Labrador Shelf

B Distribution Files

The following files constitute a complete distribution of NWAtcdbv1.0.

```
build_database
NWAtcdb.pl
NWAtcdb.db
examples/AO1987
examples/AO1987.db
examples/AO1987aval3.db
examples/aval3.db
examples/aval3.txt
examples/convertAO1987_2db
examples/convertaval3_2db
examples/example1a
examples/example1b
examples/example1c
examples/example1d
examples/example1e
source/AO1987
source/AO1987hibernia
source/CDRno102
source/CDRno62
source/CDRno71
source/CDRno91
source/CDRno94
source/CDRno96
source/CTRno100
source/CTRno113
source/CTRno157
source/CTRno40
source/CTRno46
source/CTRno66
source/moody_atlas.txt
source/nfld_shelf.txt
source/scotian_shelf_tidal.txt
source/unknown
util/append2db
util/convert2db
util/updatedb
```

C Perl Source Code

This section contains the source code of NWAtcdbv1.0.pl.

```
##########################################################################
#
#            MAIN PROGRAM
#
###########  DEFAULTS ##################################################

$sitename=".*";  # DEFAULT IS A WILD CARD (ALL LOCATIONS)
$minlong=-180;
$maxlong=360;
$minlat=-90;
$maxlat=90;
$area=".*";
$origin=".*";
$gmt_area=  "-R-75/-45/38/60";  # DEFAULT AREA TO BE COVERED BY GMT

$components = "M2,N2,S2,K1,O1";     # RECOGNIZABLE COMPONENTS

$tag_string="site_name,file_name,mooring_station,longitude,latitude,
area,origin,start_date,days_duration,water_depth,instrument_depth,
$components";  # RECOGNIZABLE TAGS

@list_of_all_tags=split(/\,/, $tag_string);
@tags=();
#@tags=@list_of_all_tags; # DEFAULT LIST INCLUCDES ALL RECOGNIZABLE
# TAGS
$components =~s/\,/\|/g;   # CREATES COMPONENTS REGEX

@component_list = ();
$gmt_view=0;
@gmt_tags=();
@out_string=();
@gmt_out_string=();

%comp_def = (
  "M2", "Lunar semidiurnal    : ampwe phawe ampsn phasn",
"O1", "Lunar diurnal        : ampwe phawe ampsn phasn",
"K1", "Luni-solar diurnal   : ampwe phawe ampsn phasn",
"N2", "Large lunar eliptic  : ampwe phawe ampsn phasn",
```

```
"S2", "Lunar Semidiurnal    : ampwe phawe ampsn phasn"
);



#################  OBTAINS ARGUMENTS FROM COMMAND LINE ################

# PRINTS A BRIEF SUMMARY OF OPTIONS IF NO ARGUMENTS ARE GIVEN

if (@ARGV == 0) {print "
************************************************************************
*   sumary of switches:
*        -f taglist (eg. site_name,days_duration,K1)
*  -s site_name    (seachers database by matching site string)
* -c component(s) (searches database by tidal component's name)
* -l minlong minlat maxlong maxlat
* -o origin   (eg. moody, lively)  Note: small caps and words
*                        connected with underscores.
* -a area     (eg. scotian_shelf, gulf_of_main) Note: small caps
*                        and words connected with underscores
* -v plot data in gmt
*
*        Short cuts can be used for tag names:
* site    = site_name
* file    = file_name
* mooring = mooring_station
*  long    = longitude
*  lat     = latitude
* date    = start_date
* days    = days_duration
* water   = water_depth
* instr   = instrument_depth
*                   names   = comp_names
* Example:      ./Tidal.pl -f site,file,lat,long,water,instr,days,S2
*                < database > output
*
************************************************************************
"; exit ();

}

# BREAKS DOWN THE ARGUMENT LIST INTO OPTIONS AND FIELDS
```

```
for($i=0;$i<@ARGV;$i++) {

$_=$ARGV[$i];
if(/^\-/) {

if(/^\-s$/) {  # site name regex: only site names
                               # containing substring are outputed
$sitename=$ARGV[$i+1];
}

if(/^\-l$/) {  # lat long bounds
                            # also used to change view size in gmt
$minlong=$ARGV[$i+1];
$minlat=$ARGV[$i+2];
$maxlong=$ARGV[$i+3];
$maxlat=$ARGV[$i+4];
                    $gmt_area=   "-R$minlong/$maxlong/$minlat/$maxlat";
}

if(/^\-o$/) {   # origin regex: only sites containing
                                # substring in origin tag are outputed
$origin=$ARGV[$i+1];
}

if(/^\-a$/) {   # area regex: only sites containing
                                # substring in area tag are outputed
$area=$ARGV[$i+1];
}

if(/^\-v$/) { # gmt view switch 1=regular view
$gmt_view=1;
@gmt_tags=("longitude","latitude","site_name");
}

if(/^\-vb$/) { # gmt view switch 2=bathymetry view
$gmt_view=2;
@gmt_tags=("longitude","latitude","site_name");
}

if(/^\-c/) { # expects comma separated list
# only data containg given component will
```

```
                         # be outputed
@component_list=split(/\,/, $ARGV[$i+1]);
}

if(/^\-f/) { # expects comma separated list
# Fields to include
@tagsp=split(/\,/, $ARGV[$i+1]);

for($k=0,$j=0; $k<@tagsp;$k++,$j++) {
$form[$j]=0;
if ($tagsp[$k] =~ m/\:/){
($tagsp[$k],$form[$j])=split(/\:/,
                                                  $tagsp[$k], 2 );
}
if ($tagsp[$k]=~m/^$components$/){
$form[$j+1]=$form[$j];
$form[$j+2]=$form[$j];
$form[$j+3]=$form[$j];
$j=$j+3;
}
}
@tags=@tagsp;
foreach $tags (@tags) {
if    ($tags eq "lat")
{$tags="latitude";}
elsif ($tags eq "long")
{$tags="longitude";}
elsif ($tags eq "mooring")
{$tags="mooring_station";}
elsif ($tags eq "site")
{$tags="site_name";}
elsif ($tags eq "file")
{$tags="file_name";}
elsif ($tags eq "date")
{$tags="start_date";}
elsif ($tags eq "days")
{$tags="days_duration";}
elsif ($tags eq "file")
{$tags="file_name";}
elsif ($tags eq "water")
{$tags="water_depth";}
elsif ($tags eq "instr")
```

```
{$tags="instrument_depth";}
elsif ($tags eq "names")
{$tags="comp_names";}
}
}
}
}


########################## MAIN READ BLOCK #############################

$source="STDIN";   # SOURCE OF DATA
$line  =<$source>; # READS FIRST LINE OF DATABASE
$line_count=0;     # COUNTER FOR OUTPUT LINES

# MAIN LOOP TO READ DATA

while( defined($line)) {

if ($line =~m/site_name/ && defined($line) ){

$data="";
chomp($line);
        $data = "$data$line\n";
                $line  =<$source>;

# READ UNTIL NEXT SITE OR EOF

while($line !~ m/site_name/ && defined($line)){

# GETS RID OF EMPTY LINES AND COMMENTS

while(($line !~ m/\w|&/ || $line =~ m/#/ )&&
defined($line) ){
$line=<$source>;
}

# APPENDS TO DATA AND READS NEXT LINE

if ($line !~ m/site_name/  &&  defined($line)) {
chomp($line);
```

```
$data = "$data$line\n";
$line  =<$source>;
}

}

# ONCE COMPLETE SITE DATA HAS BEEN READ IT'S BROKEN
# DOWN INTO SMALLER PARTS

chomp($data);
@darray = split(/&\s/, $data); # SPLITS INTO SUBDATA
   # DIVIDED BY &

 # SPLITS FURTHER INTO TAG: VALUE LINES ANDS
                     # CREATES MATRIX

for($i=0; $i < @darray; $i++){
$darrayp[$i]=[ split(/\n/, $darray[$i]) ];
}

# PREPARS HASH TABLE BY SETTING ALL TAGS TO TAG
                    # NAME

foreach $tag (@list_of_all_tags){
if($tag=~m/^$components$/) {
   $dhash{$tag} = "NA NA NA NA";
}
else {$dhash{$tag} = $tag;}
}

foreach $tag (@tags){
if($tag=~m/^$components$/) {
   $dhash{$tag} = "NA NA NA NA";
}
else {$dhash{$tag} = $tag;}
}

# MAIN NESTED LOOP OVER DATA MATRIX

for($j=0; $j < @darray;  $j++){
```

```
# CREATES HASH TABLE FROM ALL ITEMS FOUND IN
# DATABASE

$comp_names="";
$tdim= @{$darrayp[$j]};

for($i=0; $i <  $tdim; $i++){

($f1, $f2)=split(/\:\s*/,${$darrayp[$j]}[$i],
 2 );
   $f1=~s/\s+//g;

if($f1=~m/^$components$/) {
@f2_array =split(/\s+/,$f2) ;
$f2 = join(' ',@f2_array); # CLEANS UP
   # COMPONETS
$comp_names="$comp_names$f1 ";
                                      # KEEPS TRACK OF COMP. NAMES
}

$dhash{$f1}=$f2;
                             # DATA IS ENTERED INTO HASH TABLE
}

@comp_names_list=split(/\s+/,$comp_names);
@comp_names_list=sort(@comp_names_list);
$comp_names=join('-',@comp_names_list);
$dhash{comp_names}=$comp_names;
                            # LIST OF COMPONENTS IS ENTERED INTO TABLE


# CREATES OUTPUT FOR DESIRED TAGS AFTER CHECKING CONDITIONS

   $tag_switch=1; # SWITCH TO DETERMINE IF ALL
# CRITERIA IS PASSED


foreach $comp (@component_list){
 # CHECKS THAT ALL COMPONENTS SELECTED WITH -c
 # ARE PRESENT
           if ($dhash{$comp}  eq  "NA NA NA NA")
                               {$tag_switch=0;}
```

```
}

if($dhash{site_name} !~ m/$sitename/) {
$tag_switch=0;
}
elsif($dhash{latitude}>$maxlat || $dhash{latitude}
                                        <$minlat){
$tag_switch=0;
}

elsif($dhash{longitude}>$maxlong ||$dhash{longitude}
                                                   <$minlong){
$tag_switch=0;
}
elsif($dhash{area} !~ m/$area/){
$tag_switch=0;
}
elsif($dhash{origin} !~ m/$origin/){
$tag_switch=0;
}

# NOW IF THE SITE HAS BEEN CLEARED OUTPUT IS
# CREATED

if ($tag_switch) {

foreach $tag (@tags){
$out_string[$line_count] .="$dhash{$tag} ";
}

# CREATES OUTPUT FOR GMT PLOTING

foreach $tag (@gmt_tags) {
$gmt_out_string[$line_count]
.="$dhash{$tag} ";
}

$line_count++;

}

}  # end for
```

```
}    #  end if
else {$line  =<$source>; }

}  #end while


############################# WRITES OUTPUT  ###########################



# INORDER TO PRINT FORMATED DATA THE BLOCK BELOW FINDS
# MAX LENGTH OF EACH COLUMN

@line_array=split(/\s+/,$out_string[0]);
$line_len=@line_array;
for($i=0;$i<$line_len;$i++){
$max_len[$i]=length($line_array[$i]);
}

for($j=1;$j<@out_string;$j++){
@line_array=split(/\s+/,$out_string[$j]);
for($i=0;$i<$line_len;$i++){
$len_list[$i]=length($line_array[$i]);
if($len_list[$i] > $max_len[$i]){
$max_len[$i]=$len_list[$i];
}
}
}

for($i=0;$i<$line_len;$i++){
if ($form[$i] == 0){
$form[$i]=$max_len[$i]
}
}

# CREATES HEADER
$header="";
for($i=0; $i<@tags;$i++) {

if($tags[$i]=~/^$components$/) {
$header.= "
```

```
$tags[$i]ampu $tags[$i]phau $tags[$i]ampv $tags[$i]phav ";

}
else {$header .= "$tags[$i] ";}
}



#  PRINTS RESULTS TO STDOUT

$s="                                                      ";
@out_string=sort(@out_string);
unshift(@out_string,$header);
$old_line = "xxxx";

foreach $line (@out_string){
if ($line ne $old_line) {
@line_array=split(/\s+/,$line);
$old_line=$line;
for($i=0;$i<@line_array;$i++){   #FORMATS DATA
$line_array[$i]
                       =substr($line_array[$i].$s,0,$form[$i]);
}
$line=join(' ', @line_array);
print "$line\n";


}
}


# PLOTS RESULTS WITH GMT

if ($gmt_view) {
# CREATES MAP.XY FILE WITH LONG,LATS FOR PLOTING

open(gmt_map,'>map.xy');
@gmt_out_string=sort(@gmt_out_string);
$old_line = "xxxx";
foreach $line (@gmt_out_string){
if ($line ne $old_line) {print gmt_map "$line \n";}
$old_line=$line;
```

```
}
close gmt_map;

# SOME GMT OPTIONS
$point="-Sc0.1";
$triangle="-St0.15";
$square="-Ss0.1";
$proj=  "-JM6i";
$red=    "-G200/0/0";
$black= "-G0";

# REGULAR PLOT, SHOWS PLOT WITH GV, CREATES POSTSCRIPT
        # map.ps FILE AS WELL AS SITE POSTION
# FILE map.xy.

if ($gmt_view==1) {

system("pscoast $gmt_area $proj -G0/255/0 -W0.25.p
-B5 -V -K -P > map2.ps") ;# LAND
         system("psxy map.xy $gmt_area  $proj $red  $point
-O -P >> map2.ps");      # SITES
        system("ps2ps map2.ps map.ps"); # POSTSCRIPT
# FILE IS ADJUSTED
system("rm map2.ps");
system("gv -scale 2 map.ps");   # CALLS GHOSTVIEW

}


}

########################### END ##################################
```