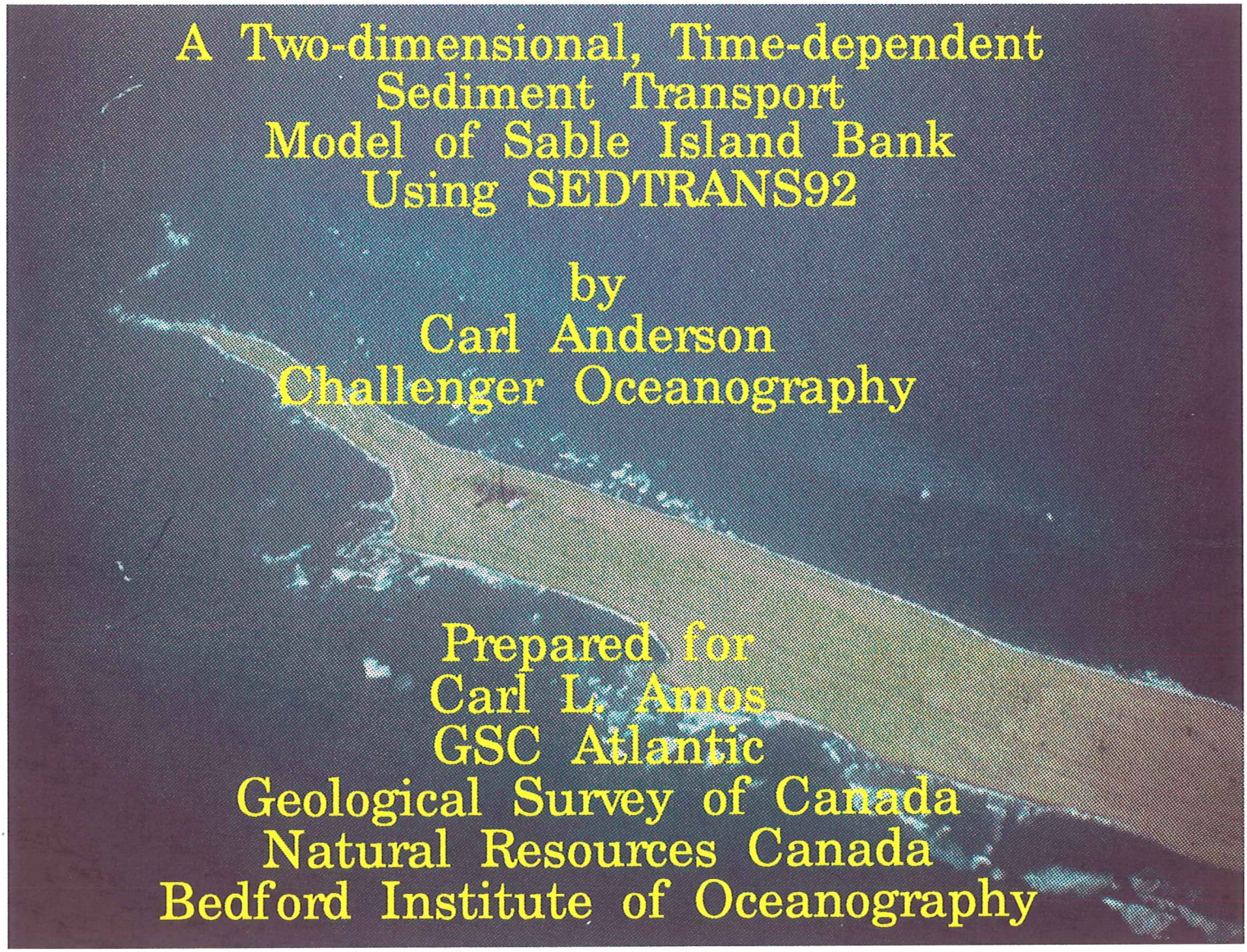
Natural Resources Canada | Ressources naturelles Canada



A Two-dimensional, Time-dependent Sediment Transport Model of Sable Island Bank Using SEDTRANS92

by
Carl Anderson
Challenger Oceanography

Prepared for
Carl L. Amos
GSC Atlantic
Geological Survey of Canada
Natural Resources Canada
Bedford Institute of Oceanography

Consultant's Final Report

25 October 1995

G.S.C. Open File Report # 2359

A TWO-DIMENSIONAL, TIME-DEPENDENT SEDIMENT TRANSPORT MODEL OF SABLE ISLAND BANK USING SEDTRANS92

by

Carl Anderson

Challenger Oceanography
25 Lawnsdale Dr.
Dartmouth, N.S., B3A 2N1

Prepared for

Carl L. Amos
GSC Atlantic
Geological Survey of Canada
Natural Resources Canada
Bedford Institute of Oceanography
P.O. Box 1006
Dartmouth, N.S., B2Y 4A2

Contractor's Final Report

25 October 1995

Geological Survey of Canada Open File Report # 2359

A TWO-DIMENSIONAL, TIME-DEPENDENT SEDIMENT TRANSPORT
MODEL OF SABLE ISLAND BANK
USING SEDTRANS92

Table of Contents

A TWO-DIMENSIONAL, TIME-DEPENDENT SEDIMENT TRANSPORT
MODEL OF SABLE ISLAND BANK
USING SEDTRANS92

LIST OF FIGURES

(Figures follow page 18)

A TWO-DIMENSIONAL, TIME-DEPENDENT SEDIMENT TRANSPORT
MODEL OF SABLE ISLAND BANK
USING SEDTRANS92

APPENDICES

A TWO-DIMENSIONAL, TIME-DEPENDENT SEDIMENT TRANSPORT MODEL OF SABLE ISLAND BANK USING SEDTRANS92

Executive Summary

A two-dimensional, time-dependent sediment transport model, SED94, incorporates the previously developed one-dimensional sediment transport model SEDTRANS92 (Li and Amos, 1993). SED94 permits SEDTRANS92 to be applied over a grid of points covering a region of interest, at successive points in time. All relevant forcing is included in the model: surface waves, tidal currents, wind-driven currents, and currents due to the general circulation. Thus, SEDTRANS92 can now be employed to investigate the effects of the passage of a storm on the mobility and transport paths of bottom sediment.

The model is used to simulate the sediment transport on Sable Island Bank during an intense late winter storm. In this case, the so-called "Storm of the Century," (14-16 March 1993) is simulated using a 1000 x 1000 m grid. Input data fields include available bathymetry, sediment grain size distribution, surface wave fields, and wind-driven and tidal current fields. Input data fields and model results are presented graphically every 6 hours during the 72-hour passage of the storm.

Detailed instructions are given for running SED94, and complete flow charts and source code are included.

A TWO-DIMENSIONAL, TIME-DEPENDENT SEDIMENT TRANSPORT MODEL OF SABLE ISLAND BANK USING SEDTRANS92

1. Introduction

The stability of the seabed with respect to erosion by waves and currents is of major importance in many offshore engineering projects, including the routing of undersea cables, and the design of undersea oil and gas pipelines. In pipeline design, the primary concern is that the pipe not become unsupported anywhere along its length as a result of erosion of the sea bottom due to currents induced by tides, wind, or surface waves.

Seabed stability is principally a function of the properties of the surficial sediment and the shear stress at the seabed. Predicting sediment transport at a given location, therefore, requires (1) the specification of the surficial sediment, (2) an evaluation of the combined seabed shear stress due to surface waves, wind-driven currents, tidal currents, and the general circulation, and (3) the application of a law relating the magnitude and direction of transport of the given sediment to the applied shear stress.

Recently, Li and Amos (1993) described a computer program, SEDTRANS92, that performs a sophisticated calculation of the time-dependent bottom shear stress over one period of the dominant surface wave, and estimates potential sediment transport using any one of several available transport algorithms. The output of the program is a point-estimate of sediment transport magnitude and direction, integrated over one wave period, for a given set of input parameters (sediment properties, water depth, surface wave magnitude, direction, and period, and background current).

To be useful in regional studies, a second program, SED94, has been developed that performs a two-dimensional, time-dependent sediment transport simulation, applying SEDTRANS92 at each point on a rectangular grid at successive points in time. A finite difference scheme accounts for the net erosion or depletion of sediment material at each grid point as a function of time. Thus, the effect of a passing storm on the bottom sediment can be assessed over a region of interest for which the bottom topography and distribution of bottom sediment properties are known.

Section 2 of this report describes a sediment transport simulation conducted with SED94 for Sable Island Bank, using input wave and wind fields from an intense late winter storm with an estimated return period of 100 years. Section 3 describes SED94 and provides instructions for its use. References are contained in Section 4, and flow charts and source code for SED94 are contained in the appendices.

2. Sediment Transport Simulation

A sediment transport simulation was conducted for Sable Island Bank using SED94 under Research and Development contract 23240-4-M026/01-OS, "A Finite Difference Scheme for SEDTRANS92- The AGC Sediment Transport Model."

2.1 Model Definition

a. Model grid

Sable Island, N.S. (44°N, 60°W) lies on the eastern edge of UTM Zone 20

(C.M. 63°W), or the western edge of UTM Zone 21 (C.M. 57°W). The Sable Island Bank grid for SED94 is referenced to the Universal Transverse Mercator (UTM) grid, zone 20 (x = 500.0 km at the central meridian of 63°W, and y = 0 km at the equator). The domain is a 253 km x 190 km rectangle with 567.0 <= x <= 820.0 km, and 4,750.0 <= y <= 4,940.0 km in the UTM grid. It is divided into 48,070 1 km x 1 km cells, in 190 rows and 253 columns. The centres of the cells thus have coordinates 567.5 km x (i - 1), and 4,750.5 km x (j - 1) km, where 1 <= i <= 253 and 1 <= j <= 190 are the cell coordinates. This grid is referred to hereafter as the "Sable Island Bank U1 grid", or simply as the "U1 grid". The bounds of the grid are as follows:

Corner	x (m)	y (m)	Longitude (W)	Latitude (N)
SW	567000	4750000	62° 10′ 45.595"	42° 54′ 05.775"
SE	820000	4750000	59° 05′ 04.797"	42° 50′ 15.161"
NW	576000	4940000	62° 02′ 31.931"	44° 36′ 40.630"
NE	820000	4940000	58° 58′ 18.687"	44° 32′ 39.147"

b. Bathymetry

High-resolution bathymetric data were provided by Bob Courtney, GSC Atlantic, for the U1 grid. The positions of the high-resolution depth soundings correspond to the centres of the grid cells. Wherever high-resolution depth data were unavailable, low-resolution (4 x 4 minute) Scotian Shelf bathymetric data were used. The low-resolution data set was originally created by Dave Greenberg, of DFO, for numerical modelling on the Scotian Shelf. First, the low-resolution data were bi-linearly interpolated to completely fill the U1 grid. The available high-resolution depth data were then substituted for the low-resolution data. Finally, any depths greater than 1000.0 m were assigned a value of 999.9 m.

Figure 1 is a contour map of Sable Island bathymetry (depth in metres) over the model grid.

c. Surficial sediment grain size

The U1 surficial sediment grain size was obtained by gridding the grain size distribution depicted in Figure 4 of Amos and Nadeau (1988). The original grain size versus location data were not available; therefore, the grain size contours in Figure 4 of Amos and Nadeau (1988) were digitized by John Zevenhuizen, of Orca Marine. These data were then gridded using the Barnes algorithm (Daley, 1991) in which weighted averages of scattered observations are assigned to each grid position. A contour plot of the gridded grain size showed the contours of Amos and Nadeau's Figure 4 to be well reproduced.

Figure 2 is a contour map of Sable Island surficial sediment grain size (microns) over the model grid.

2.2 Oceanographic Input Fields

a. Surface Wave field

The 14-16 March 1993 East Coast winter storm, known as the "Storm of the Century" (Cardone, *et al*, 1994), was used to represent a 100-year return period storm event on the Scotian Shelf. Surface wind and wave fields for

this storm were mapped onto the model grid at 3 hour intervals.

Wave fields for the 14-16 March 1993 storm were hindcast by Bash Toulany and Will Perry, of DFO, using the "Wave Watch" third-generation wave forecast model (Gunther *et al*, 1992). This wave model includes the effects of shallow water on the phase speed, dissipation, and refraction of wind-driven surface waves, but does not include the effect of tidal currents. The wave field parameters obtained from the model were those required by SED94: (1) the significant wave height, (2) average wave period, and (3) average wave direction.

The hindcast was performed in two steps. First, the wave model was run over a 140 x 140 cell grid covering the North Atlantic ocean with a resolution of 50 km, referred to as the "R50", or "coarse" grid. The second step was to force a 9 x 8 cell subgrid model with a subset of the R50 model output. The nested subgrid covered the eastern portion of the Scotian Shelf at a resolution of 10 km, and was referred to as the "R10", or "fine" grid. The output wave fields of the R10 grid were then bi-linearly interpolated every three hours onto the U1 model grid. The effects of bottom topography were taken into account in both the coarse and fine grid wave model runs.

Figures 3.1- 3.12 depict the hindcast Sable Island Bank wave field at six-hour intervals during the passage of the Storm of the Century. The arrows point in the direction of surface wave advance, and are proportional in length to the wave height. For the sake of clarity, arrows are plotted only for every fifth grid point.

b. Tidal current

Hourly tidal current maps were generated for the 72-hour storm period. Five tidal constituents were included in the tidal current prediction. The K1, O1, M2, Scotian Shelf tidal current component amplitudes and phases, determined from numerical simulations by deMargerie and Lank (1986) on a 4 x 4 minute grid, were provided by Doug Gregory, DFO. The N2 and S2 amplitudes and phases were inferred from M2 in accordance with deMargerie and Lank (1986), and the five sets of current amplitudes and phases were then bi-linearly interpolated onto the U1 grid. Tidal current ellipse parameters were then computed and used to compute the hourly tidal currents.

Figures 4.1- 4.9 are plots of the hindcast tidal current vectors every three hours at every fifth U1 grid point over a 24-hour period during the Storm of the Century.

c. Wind-driven current

Maps of the U1 surface wind field during the Storm of the Century were generated from the wave forecast model wind fields for use in hindcasting the wind-driven component of the current. Figures 5.1- 5.12 depict the surface wind vectors at every fifth U1 grid point for the hours corresponding to the surface wave fields of Figs. 3.1- 3.12.

For the purposes of this simulation, the wind-driven current was taken as being the Ekman flux divided by the depth, limited to 3 percent of the wind speed. Surface wind stress was calculated from the wind velocity following Smith (1980).

d. Background current

No suitable digitized map of the general circulation over Sable Island Bank was available for use in this simulation. Therefore, the background current was set to zero. In view of the extreme intensity of the wind and wave fields, it was judged that the contribution of the general circulation to the sediment transport would be negligible during the Storm of the Century. It should, however, be included in simulations of less intense storms.

2.3 Results

a. Sediment transport

Figures 6.1- 6.13 show the six-hourly sediment transport vectors resulting from the oceanographic forcing shown in Figures 3, 4, and 5 for the Storm of the Century. The vector lengths are scaled logarithmically, and only every fifth vector is plotted. The sediment transport magnitude over the model domain is contoured in Figures 7.1- 7.13. Contours are drawn for transport magnitudes of 10^{-n} m^2s^{-1} (cubic meters of sediment meter of seabed width per second), where $n < 9$.

b. Sediment erosion/deposition

Six-hourly contour maps of total accumulated sediment are presented in Figures 8.1- 8.13. Positive values of accumulated sediment correspond to net deposition; negative values, to net erosion. Contours are drawn for $\pm 10^{-n}$ m (meters of accumulated sediment), where $n < 5$.

3. Sediment Transport Model SED94

3.1 Background

The computer program SEDTRANS92 (Li and Amos, 1993) is a one-dimensional sediment transport model that estimates the mass and volume transport of sediment per unit width of seabed for specified water depth, sediment grain size, ambient current, and surface wave field. Any one of seven different transport algorithms may be selected by the user (see Li and Amos, 1993).

The following diagram illustrates the application of program SEDTRANS92 to a single point (x,y) at time (t):

```
Water          Surficial sediment     Significant wave          Current
depth             grain size          Ht., dir., per.        Mag., dir.,z
(x,y)              (x,y)                (x,y,t)                 (x,y,t)
  |                  |                     |                       |
  +------------------+-----------+---------+----------------------+
                                |
                            SEDTRANS92
                                |
                      Sediment transport
                           (x,y,t)
```

Program SED94 is a two-dimensional, time-dependent sediment transport model that applies SEDTRANS92 over a rectangular grid of points over which water depth, sediment grain size, background current, tidal current, wind-driven current, and surface waves are defined. SED94 was written and

tested in 1994 by Challenger Oceanography, under Research and Development contract 23240-4-M026/01-OS, "A Finite Difference Scheme for SEDTRANS92- The AGC Sediment Transport Model". Program modifications were made in 1995 by Challenger Oceanography under Contract 23420-4-M424/01-HAL "Oceanographic Support to the AGC Finite Difference Sediment Transport Model".

In SED94, SEDTRANS92 is called as a subroutine at each grid point and at each time step. The modifications to SEDTRANS92 for this application are limited to the input/output (I/O) routines; the physics remain the same.

Program SED94 may be run for a single time step to evaluate sediment transport over the grid for a given time-independent set of conditions. Alternatively, when maps of the current and wave fields are available for successive times separated by a constant time interval, SED94 estimates sediment transport over the model domain at those times, and keeps a running account of the accumulation or depletion of sediment at each grid point. Maps of sediment transport magnitude and direction, sediment accumulation, and the occurrence of wave breaking are output at desired time intervals.

The following diagram illustrates the operation of SED94. Input and output quantities are defined over the model grid for one or more times separated by a constant time interval:

```
Bathymetry  Surficial sediment  Surface Wave  Background  Tidal   Wind-driven
    |           grain size        field        current   current    current
    |               |               |             |        |           |
    +---------------+---------------+-------------+--------+-----------+
                                    |
                                  SED94
                                    |
                +-------------------+-------------------+
                |                   |                   |
            Sediment                                  Sediment
            transport                           accumulation/erosion
```

3.2 General Procedure

The general procedure for running SED94 is to:

(a) define the model grid boundaries and spatial resolution, using program SETGRID,

(b) assign model run parameters such as start date and time and run duration, using program SETUP,

(c) define physical parameters such as roughness length, water and sediment densities, etc., in file 'sed94.par',

(d) assemble the required input maps, and

(e) link and run SED94.

These steps are summarized in the following flow chart, in which parentheses () denote programs, and square brackets [] denote disk files:

```
                 Interactive
                  user input
                      |
                 (SETGRID)
                      |            Interactive
                [setgrid.out]       user input
                      |                  |
                      +--------+--------+         maps of:
                               |                    [bathymetry]
                            (SETUP)                 [grain size]
                               |                    [background current]
                      +-----------------+           [tidal currents]
                      |                 |           [wind-driven currents]
[sed94.par]     [sed94.in]      [sed94.inc]         [wave field]
     |                |                 |                  |
     +----------------+-----------------+-----------------+
                                        |
                                     (SED94)
                                        |
              +---------------+-----------------+
              |               |                 |
           [names]       [sed94.log]        maps of:
                                         [sediment transport]
                                         [accumulated sediment]
                                         [wave breaking]
```

3.3 Preliminary Steps

a. Defining the model grid

SETGRID is an interactive program that establishes the model grid. The user is prompted to enter the desired grid boundaries and spatial resolution. The program then prints a map of the selected grid to the terminal screen.

SED94 uses a Universal Transverse Mercator (UTM) grid. UTM "zones" are 6° wide meridional strips of the earth's surface centred on a Central meridian (C.M.), and extending from -90° (the South Pole) to $+90^{\circ}$ (the North Pole). UTM coordinates are x and y, measured in meters. x = 500,000 m at the C.M. of each zone, and y = 0 at the Equator.

Program SETGRID is run by issuing the command "setgrid" from the "setup" subdirectory. SETGRID asks the user to enter the following grid parameters:

1.	izone	UTM zone
2.	xmin	UTM x coordinate (m) at SW corner of grid
3.	ymin	UTM y coordinate (m) at SW corner of grid
4.	xmax	UTM x coordinate (m) at NE corner of grid
5.	ymax	UTM y coordinate (m) at NE corner of grid
6.	resolution	dx = dy (m)

The program calculates and displays the remaining grid parameters:

7.	ncols	number of columns in the grid
8.	nrows	number of rows in the grid
9.	xspan	West-East dimension of grid (meters)
10.	yspan	South-North dimension of grid (meters)
11.	npts	number of grid points

The grid parameters are also written to file 'setgrid.out', which contains the following parameters on one line:

xmin ymin dx dy ncols nrows izone

'setgrid.out' is an input file to program SETUP, described in the next section.

b. Assigning model run parameters

SETUP is an interactive program that prompts the user for information used to control the model run. SETUP is run by issuing the command "setup" from the "setup" subdirectory. The program requests the following information:

1. Run identifier (2 alphabetic characters). The identifier becomes the leading characters of the file names for (1) the run log file and (2) all output map files (Sect. 3.5).
2. Name of run (quoted string up to 75 characters long).
3. Year, month, day, and hour at the start of the model run.
4. Length of model run in hours. Length zero causes SED94 to run through only one time step.
5. Bathymetry map file path name and file name.
6. Sediment grain size distribution map file path name and file name.
7. Background current map file path name and file name.
8. Tidal current map file path name and 2-letter identifier.
9. Wind-driven current map file path name, 2-letter identifier, and time interval.
10. Wave field map file path name, 2-letter identifier, and time interval.
11. Time intervals for output maps of estimated sediment transport, accumulated sediment, and location of breaking waves.

Program SETUP reads the grid parameters from the file 'setgrid.out'. If 'setgrid.out' doesn't exist, SETUP fails. SETUP prints the setup information to the terminal and creates two files containing model information required by SED94:

1. 'sed94.inc' is an "include" file containing the Fortran parameter statement "PARAMETER (NCOLS = ncols, NROWS = nrows)" that supplies the two parameters required for all DIMENSION and DATA statements in the model source code 'sed94'.
   An example is:

PARAMETER (NCOLS = 253, NROWS = 190)

2. 'sed94.in' contains the model run parameters described above. It must be in the working directory when running SED94. Appended to 'sed94.in' is a full list of input files (full pathnames) required by the model run. The list is not read by SED94, but can be used to check whether all required map files are present before the run. Missing map files cause SED94 to fail. File 'sed94.in' can be modified to alter run name, run id, file names, paths, start time, run duration, or output file time intervals.

The following is an example of 'sed94.in':

```
Name:
'Storm of the Century                                                  '

Run ID:
'aa'

  xmin(m.)  ymin(m.)     dx(m)     dy(m)    columns      rows     izone
    567000   4750000      1000      1000       253        190        20

Startyr  Starthours   Start date/time         Runhours
93       1728        '1993 Mar 14 00:00 UT'    72

Input DT (hrs.):  Tidal current   Wind-driven current  Wave field
                     1               3                    1

Output DT (hrs.): Sed. transport  Accumulated sed.     Wave breaking
                     3               3                    3

Bathymetry map (pathname, file name):
'/usr30/anderson/bathymetry/              '  'Sable.dep          '

Grain size map (pathname, file name):
'/usr30/anderson/grain/                   '  'Sable.siz          '

Background current map (pathname, file name):
'/usr30/anderson/currents/mean/           '  'Sable.bac'         '

Tidal current maps (pathname, file ID):
'/usr30/anderson/tides/maps/              '  dm

Wind-driven current maps (pathname, file ID):
'/usr30/anderson/currents/windriven/      '  ek

Wave field maps (pathname, file ID):
'/usr30/anderson/waves/maps/              '  fg

Tidal current input files:
   /usr30/anderson/tides/maps/dm931728.tid
           through
   /usr30/anderson/tides/maps/dm931800.tid

Wind-driven current input files:
   /usr30/anderson/currents/windriven/ek931728.win
           through
   /usr30/anderson/currents/windriven/ek931800.win
Wave field input files:
   /usr30/anderson/waves/maps/fg931728.waves
           through
   /usr30/anderson/waves/maps/fg931800.waves
```

Model grid parameters and model run parameters should only be changed by re-running SETGRID and SETUP.

c. Defining physical parameters

Program SED94 reads the following physical parameters from file 'sed94.par' and passes them to subroutine 'sed92':

```
 rhos = density of sediment mineral(s)    (kg/m**3)
fract = fraction of the total sediment with specified grain size
  rkb = bottom roughness height (m)
 rhow = density of fluid (water)   (kg/m**3)
iopt1 = sediment transport predictor option number. See Li and Amos (1993)
iopt2 = friction factor predictor option number. See Li and Amos (1993)
conc0 = initial estimate of sediment concentration (ppm) (i.e., mg/l)
towce = critical stress for erosion (pa)
towcd = critical stress for deposition (pa)
   ws = settling velocity (m/s)
  prs = probability of resuspension (normally assumed = 1.0)
rkero = proportionality coefficient for erosion rate (default = 2.0)
```

An example of 'sed94.par' follows:

rkb	*fract*	*rhos*	*rhow*	*z*	*iopt1*	*iopt2*
0.03	*1.0*	*2650.0*	*1025.0*	*1.0*	*3*	*1*
conc0	*towcd*	*towce*	*ws*	*prs*	*rkero*	
1.0	*1.0*	*1.0*	*1.0*	*1.0*	*1.0*	

'sed94.par' must be in the working directory when SED94 is run.

3.4 Running SED94

After SETUP has been run, SED94 must be linked with the file 'sed94.inc'. Issue the command "make install" from the "sed94" subdirectory (see Sect. 3.6, below).

To run SED94, change the working directory to the subdirectory under '~anderson/runs/' where the desired 'sed94.in' and 'sed94.par' files reside. Then issue the command "sed94" to start the model. As SED94 runs, informative messages are displayed on the terminal and written to the log file 'ccSED94.log', where 'cc' is the run identifier from line 5 of 'sed94.in'. The log file and map files of sediment transport, accumulated sediment, and wave breaking are saved in the working directory.

The following is the log file from a short SED94 run:

aaSED94.log 14:18:56 31-Mar-95

```
 Program SED94- Sediment transport simulation using SEDTRANS92.

 Name:
 'Storm of the Century                                                    '
 Run ID:
 'aa'
 xmin(m.)      ymin(m.)    dx(m)    dy(m)    columns   rows   UTM zone
  567000      4750000       1000     1000       253          190 20
 Startyr  Starthours   Start date/time        Runhours
 93       1728         '1993 Mar 14 00:00 UT'   3
```

```
Input DT (hrs.):  Tidal current   Wind-driven current  Wave field
                   1               3                   1
Output DT (hrs.): Sed. transport  Acc. sed.  Wave breaking
                   3                 3             3
Bathymetry map (directory, file name):
'/usr30/anderson/bathymetry/              '  'Sable.dep           '
Grain size map (directory, file name):
'/usr30/anderson/grain/                   '  'Sable.siz           '
Background current map (directory, file name):
'/usr30/anderson/currents/mean/           '  'Sable.bac           '
Tidal current maps (directory, file ID):
'/usr30/anderson/tides/maps/              '  dm
Wind-driven current maps (directory, file ID):
'/usr30/anderson/currents/windriven/      '  ek
Wave field maps (directory, file ID):
'/usr30/anderson/waves/maps/              '  fg

 PHYSICAL PARAMETERS:

    rkb        fract      rhos       rhow       z          iopt1 iopt2
  3.000E-02  1.000E+00  2.650E+03  1.025E+03  1.000E+00     1     1

    conc0      towcd      towce      ws         prs        rkero
  1.000E+00  1.000E+00  1.000E+00  1.000E+00  1.000E+00  1.000E+00

   Get map from Sable.dep

Grid has  34433 sediment transport cells
              0 land cells
          13637 deep water cells
   Get map from Sable.siz
   Get map from Sable.bac

Hour = 1728. Elapsed hours =    0. 1993 Mar 14 00:00 UT
   Get map from dm931728.tid
   Get map from ek931728.win
   Get map from fg931728.waves
   Put map in aa931728.tpt
   Put map in aa931728.acc
   Put map in aa931728.brk

Hour = 1729. Elapsed hours =    1. 1993 Mar 14 01:00 UT
   Get map from dm931729.tid
   Get map from fg931729.waves

Hour = 1730. Elapsed hours =    2. 1993 Mar 14 02:00 UT
   Get map from dm931730.tid
   Get map from fg931730.waves

Hour = 1731. Elapsed hours =    3. 1993 Mar 14 03:00 UT
   Get map from dm931731.tid
   Get map from ek931731.win
   Get map from fg931731.waves
   Put map in aa931731.tpt
   Put map in aa931731.acc
   Put map in aa931731.brk
Done at 14:21:02     31-Mar-95.
```

3.5 Map Files

All gridded input and output variables in SED94 are stored in map files, each consisting of a 5-line header followed by one or more data matrices. Map files contain 2-D fields of 1-, 2-, 3-dimensional variables. Each of the variables composes a "layer" of the file. Thus, a scalar variable has one layer, a vector variable, two layers (e.g., Tx and Ty), and the wave field specification has three layers (wave height, direction, and period).

The variable map describes the spatial variation of the variable over the model domain at a given time, resolved to the nearest hour. Successive maps of the variable describe the time history of the spatial distribution of the variable. In the case of time-independent maps such as bathymetry or grain size, there is no particular time associated with the map.

SED94 uses the following map files, identified by their file name extensions:

Variable	Extension	Layers
Bathymetry	.dep	1
Grain size	.siz	1
Background current	.bac	2
Tidal current	.tid	2
Wind-driven current	.win	2
Wave field	.waves	3
Sediment transport	.tpt	2
Accumulated sediment	.acc	1
Wave breaking	.brk	2

a. Map file names

Map files that contain time-dependent variables have names of the form 'ccyyhhhh.ext', where 'cc' is a two-letter identifier, 'yy' designates the year, 'hhhh' is the number of hours since 00:00 1 January of year 'yy', and 'ext' is the file name extension denoting the variable stored in the map.

Map files containing time-independent variables have names of the form 'name.ext', where 'name' is any descriptive word. An example is 'Sable.dep', which is a bathymetry file for Sable Island Bank.

SED94 writes a list of output map file names (without extensions) in the output file 'names'. This list is used when plotting model results.

b. Headers

All map files begin with a five-line header containing the following information:

title1
title2
xmin ymin dx dy ncols nrows layers izone
timestamp
year yearhour

The following is an example of a map file header:

Storm of the Century
Sediment Transport
567000 4750000 1000 1000 253 190 3 20
1993 Mar 14 01:00 UT
93 1728

The titles (lines 1 and 2) may contain any desired descriptive information.

The date stamp gives the date and time corresponding to the year and hour of the year. The time stamp (line 4) appears on all plots made by 'gri' scripts. The year and hour of the year (line 5) are the same as those in the file name. If the mapped variable is time-independent, (e.g., depth, or grain size), header line 4 may contain a legend like "Time-independent". In that case, line 5 contains a bogus year and yearhour.

c. Data Matrices

Following the header (i.e., starting on line 6) are one or more data matrices. The data in each "layer" are arranged naturally as they would be on a map of the model domain- i.e., the first row corresponds to the northern boundary, the bottom row corresponds to the southern boundary, etc.

3.6 Maintaining the Programs

If source code is modified, or subroutines are added, the programs must be re-compiled and linked before they can be executed. Source code for new subroutines must have file name extension ".f".

Source code is written in standard Fortran using the "tab" convention, and lines longer than 72 characters. Two system subroutines are required. Compiler options "+es E1" are needed to accommodate the source code and provide access to the system subroutines. Compiling source code and linking object code is simplified by the use of the procedure 'flink' and the UNIX "make" facility.

Self-contained programs (SETGRID, SETUP) are compiled and linked using the procedure 'flink', as described below.

Subroutines and include files required by SED94 are separate files in their respective subdirectories. This permits using the UNIX "make" facility to compile and link, or "make", the program. "Making" ensures that the executable program always uses the latest version of every program unit, without re-compiling units that have not been modified since the previous "make".

Another program, 'mkmf', short for "make makefile", is used to set up the makefile.

a. Using the script 'flink'

'flink' stands for "Fortran compile and link." The executable is located in "/usr30/anderson/bin" so it can be invoked from anywhere. Issue the command "flink sourcefile" from the program subdirectory, where "sourcefile" is the the source file name. 'flink' installs the linked executable program

in "usr30/anderson/bin" so it can be invoked from any directory.

b. Using 'make'

A program is "made" according to the instructions in the file 'Makefile' in the program subdirectory along with source code files. The program is "made" by issuing the command "make install" from the program subdirectory. The "install" option copies the executable file to the subdirectory '/usr30/anderson/bin' so it can be executed from any subdirectory.

c. Using 'mkmf'

Use "mkmf" to edit file 'Makefile' if subroutines are added to a program, or to create 'Makefile' if it is accidentally deleted. Issue the command "mkmf" from the program directory. 'mkmf' edits an existing 'Makefile' by adding any new subroutines it finds in the working directory. If 'Makefile' does not exit, a new makefile is created. Modify any new 'Makefile' as follows:

1. Alter the "DEST" line to read
   DEST = */usr30/anderson/bin*

2. Add the following line after the "EXTHDRS" line:
   FFLAGS = *+es +E1*

3. Alter the "LD" line to read
   LD = *f77*

4. Alter the "PROGRAM" line to read
   PROGRAM = *sed94*

4. References

Amos, Carl L., and Odette C. Nadeau, 1988. Surficial sediments of the outer banks, Scotian Shelf, Canada. Canadian Journal of Earth Sciences, 25(12) 1923-1944.

Cardone, V.J., R.E. Jemsen, D.T. Resio, V.R. Swail, and A.T. Cox, 1994. Evaluation of contemporary wave models in rare extreme events: "Haloween Storm" of October, 1991; "Storm of the Century" of March, 1993.

Daley, Roger, 1991. Atmospheric Data Analysis. Cambridge Press, New York.

de Margerie, Sylvain, and Karen D. Lank, 1986. Tidal circulation of the Scotian Shelf and Grand Banks. ASA Consulting Ltd., Dartmouth, N.S., prepared under Contract 08SC.FP901-5-X515 for the Department of Fisheries and Oceans.

Gunther, Heinz, Susanne Hasselmann, and P.A.E.M. Janssen, 1992. The WAM Model cycle 4. Report No. 4, Deutsches KlimaRechenZentrum, Hamburg.

Li, Michael Z. and Carl L. Amos, 1993. SEDTRANS92: Re-evaluation and Upgrade of the AGC Sediment Transport Model. Geological Survey of Canada, Bedford Institute of Oceanography, Open File 2769, August 1993, 114 pp.

Smith, S.D., 1980. Wind stress and heat flux over the ocean in gale force winds. J. Phys. Oceanogr. 10, 709-726.

Sable Island Bank Bathymetry

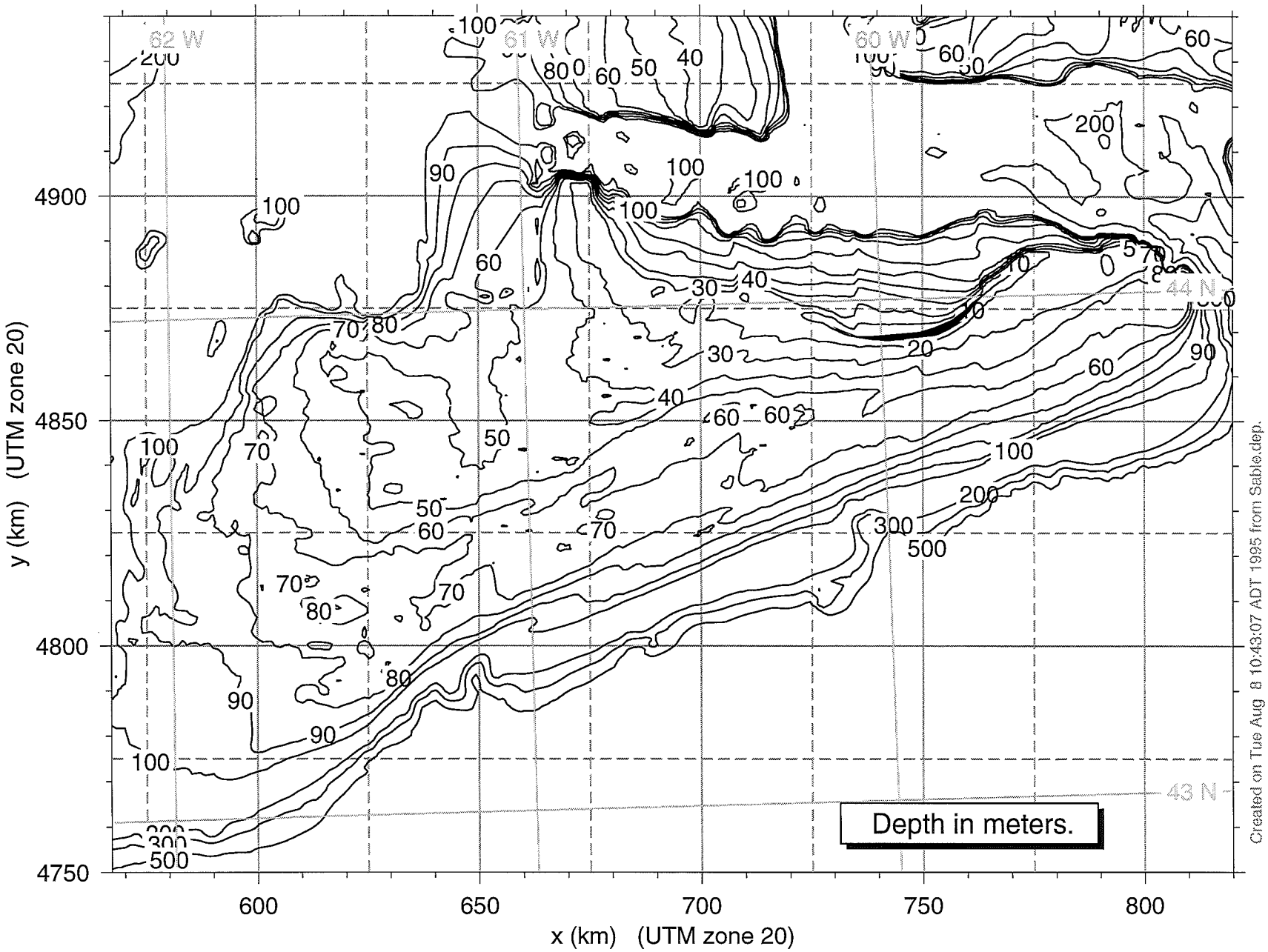


Figure 1

Sable Island Bank Surficial Sediment Grain Size

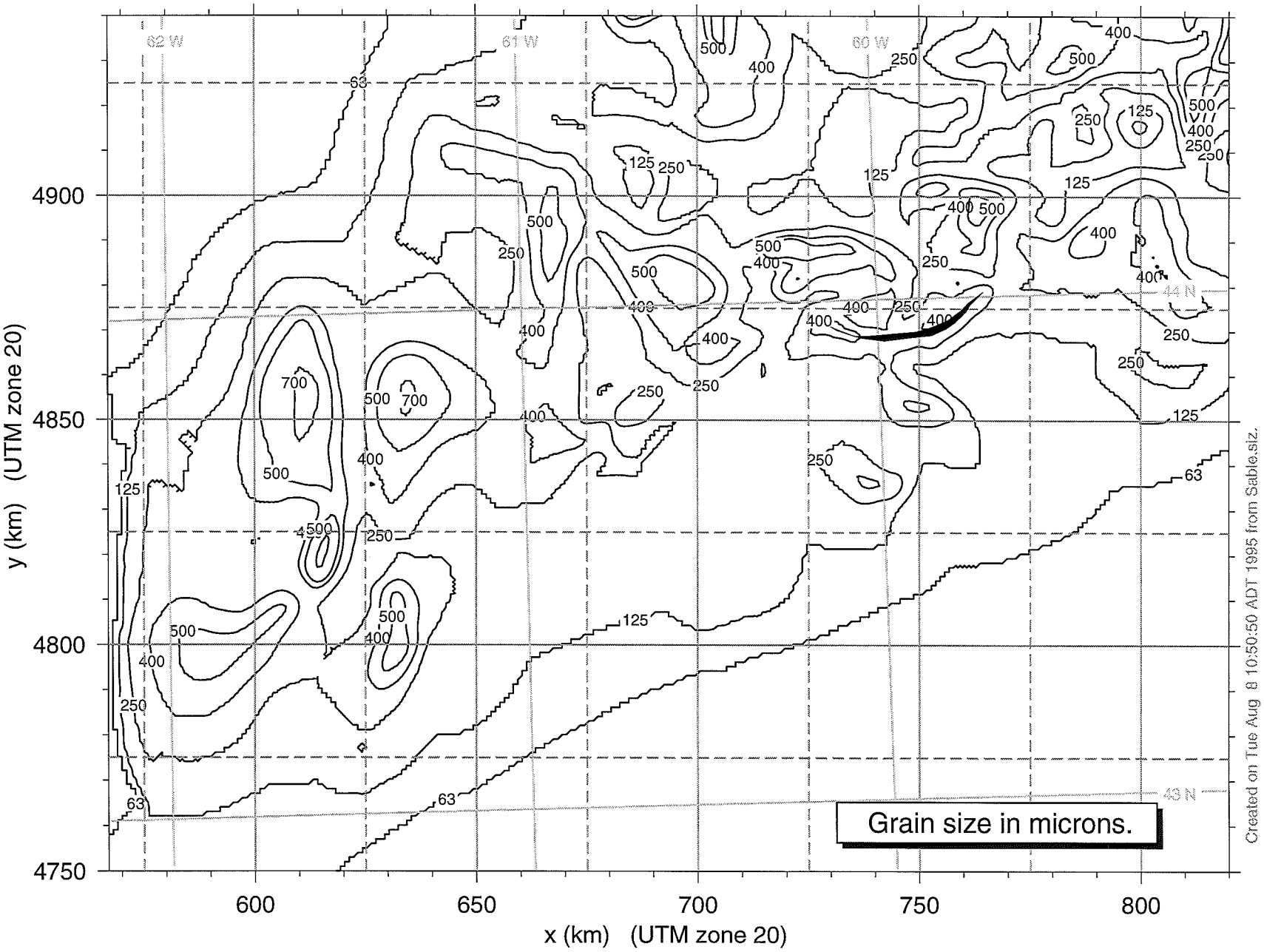


Figure 2

Sable Island Bank Surface Waves

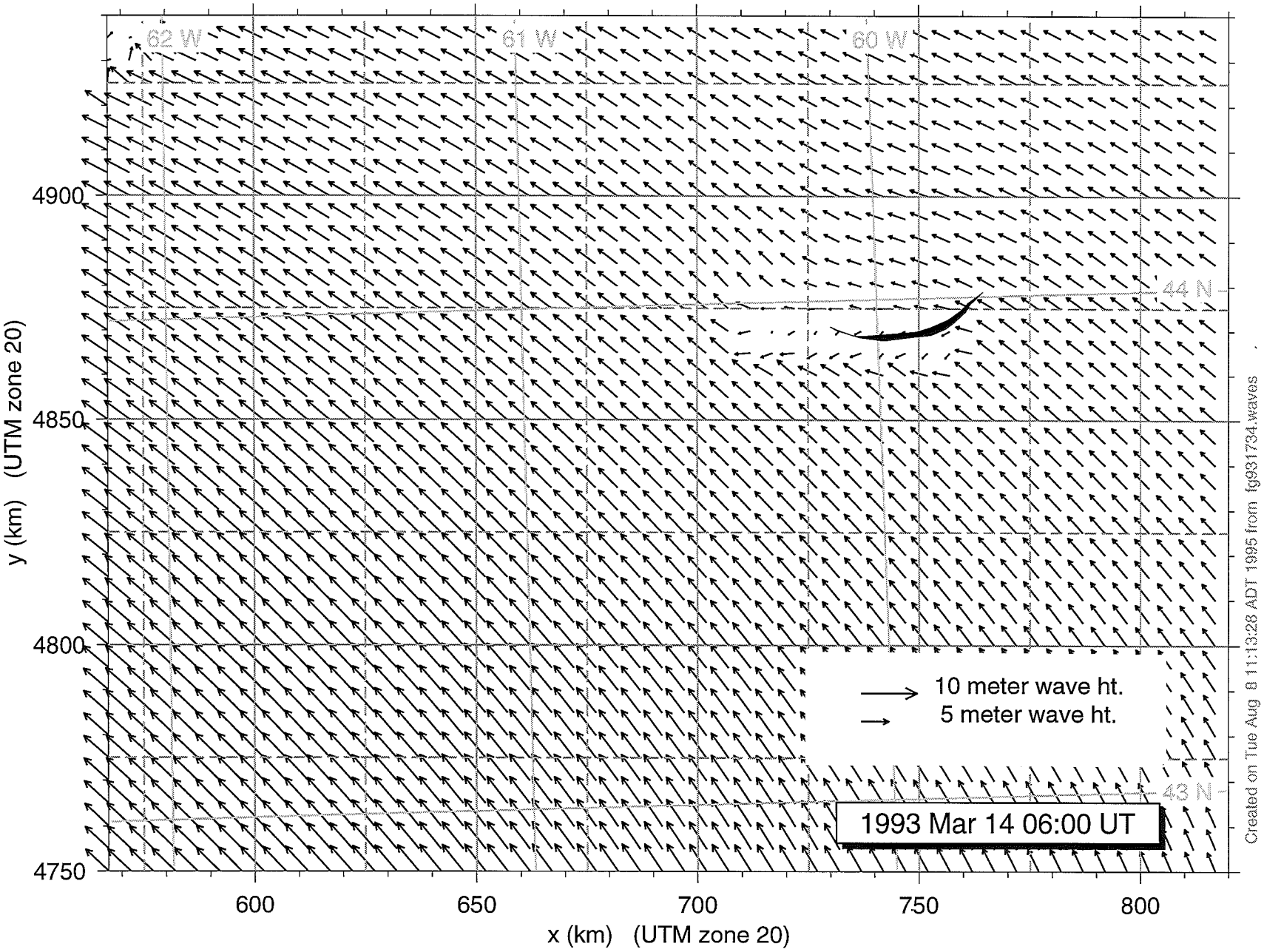


Figure 3.1

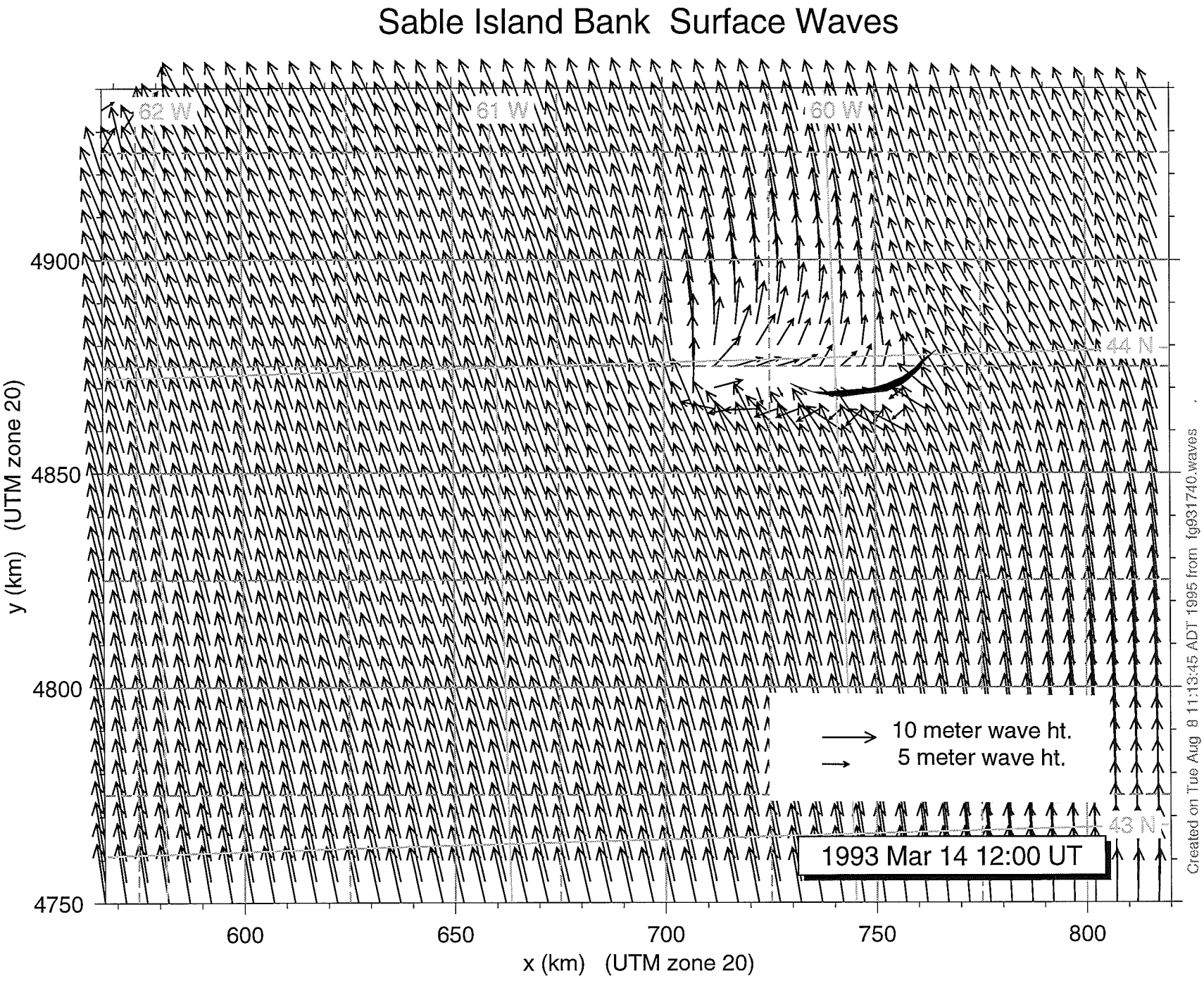

Sable Island Bank Surface Waves
62 W
61 W
60 W
44 N
43 N
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
10 meter wave ht.
5 meter wave ht.
1993 Mar 14 12:00 UT
Created on Tue Aug 8 11:13:45 ADT 1995 from fg931740.waves

Figure 3.2

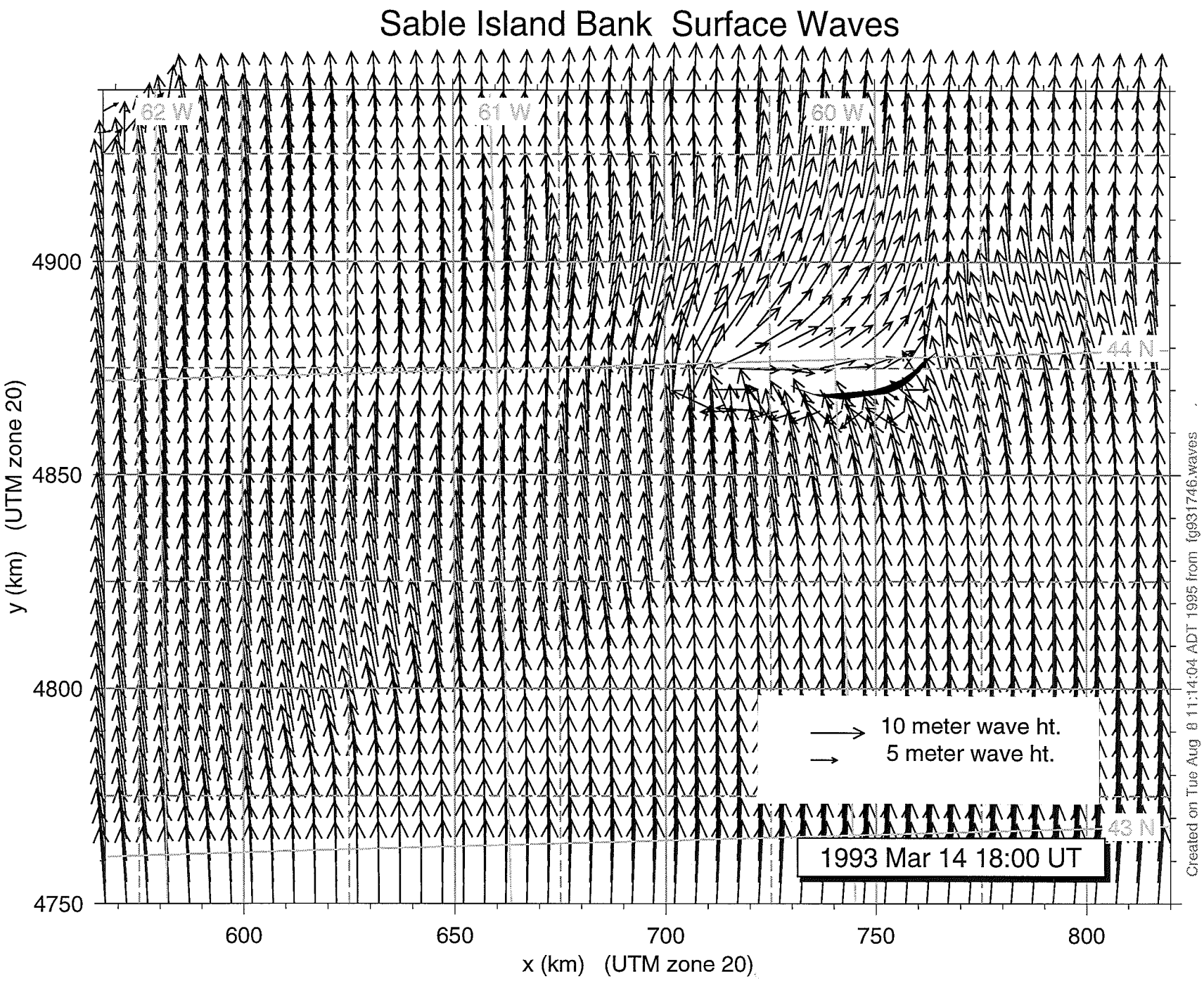

Sable Island Bank Surface Waves
62 W
61 W
60 W
44 N
43 N
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
10 meter wave ht.
5 meter wave ht.
1993 Mar 14 18:00 UT
Created on Tue Aug 8 11:14:04 ADT 1995 from fg931746.waves

Figure 3.3

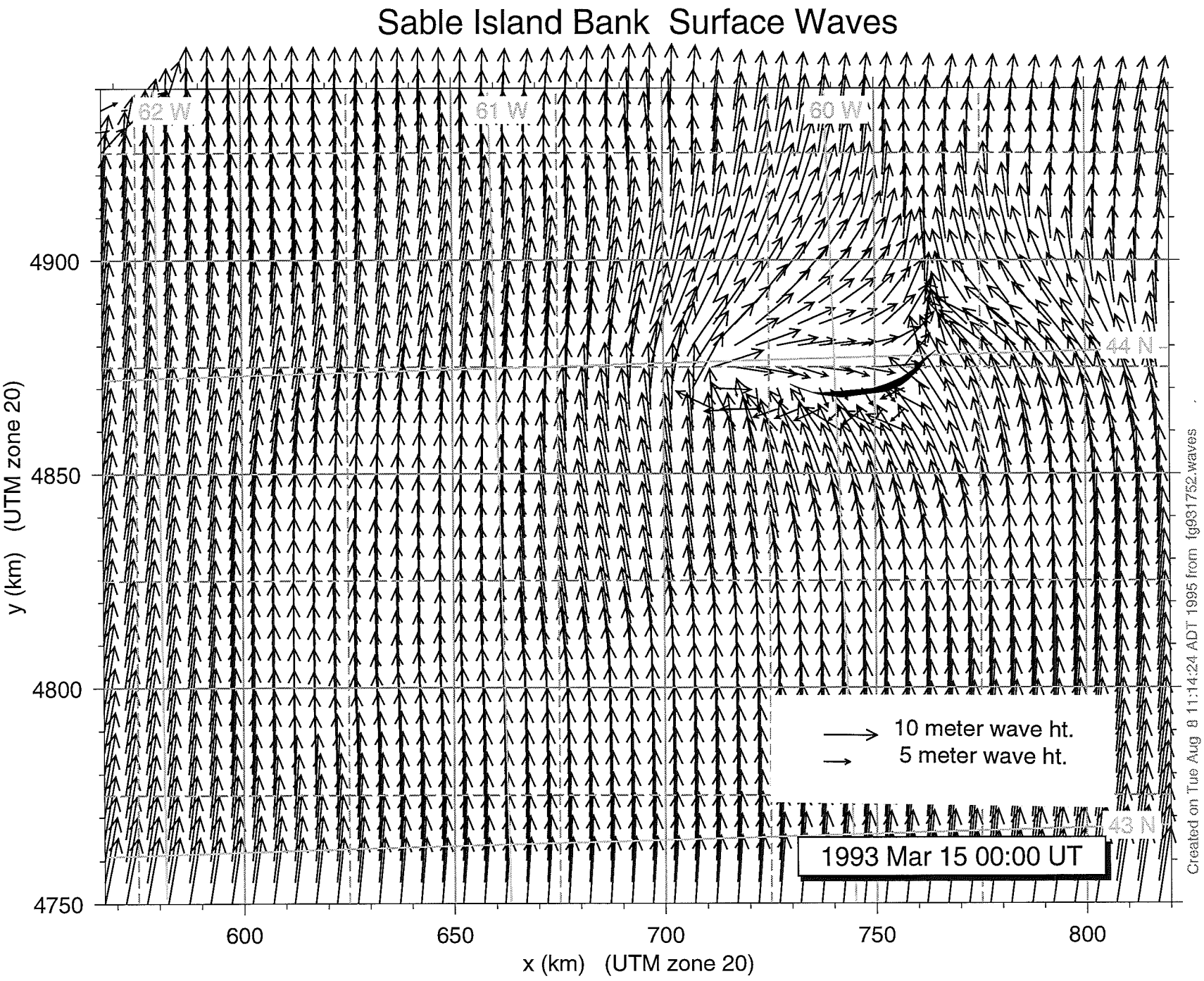

Sable Island Bank Surface Waves
62 W
61 W
60 W
44 N
43 N
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
10 meter wave ht.
5 meter wave ht.
1993 Mar 15 00:00 UT
Created on Tue Aug 8 11:14:24 ADT 1995 from fg931752.waves

Figure 3.4

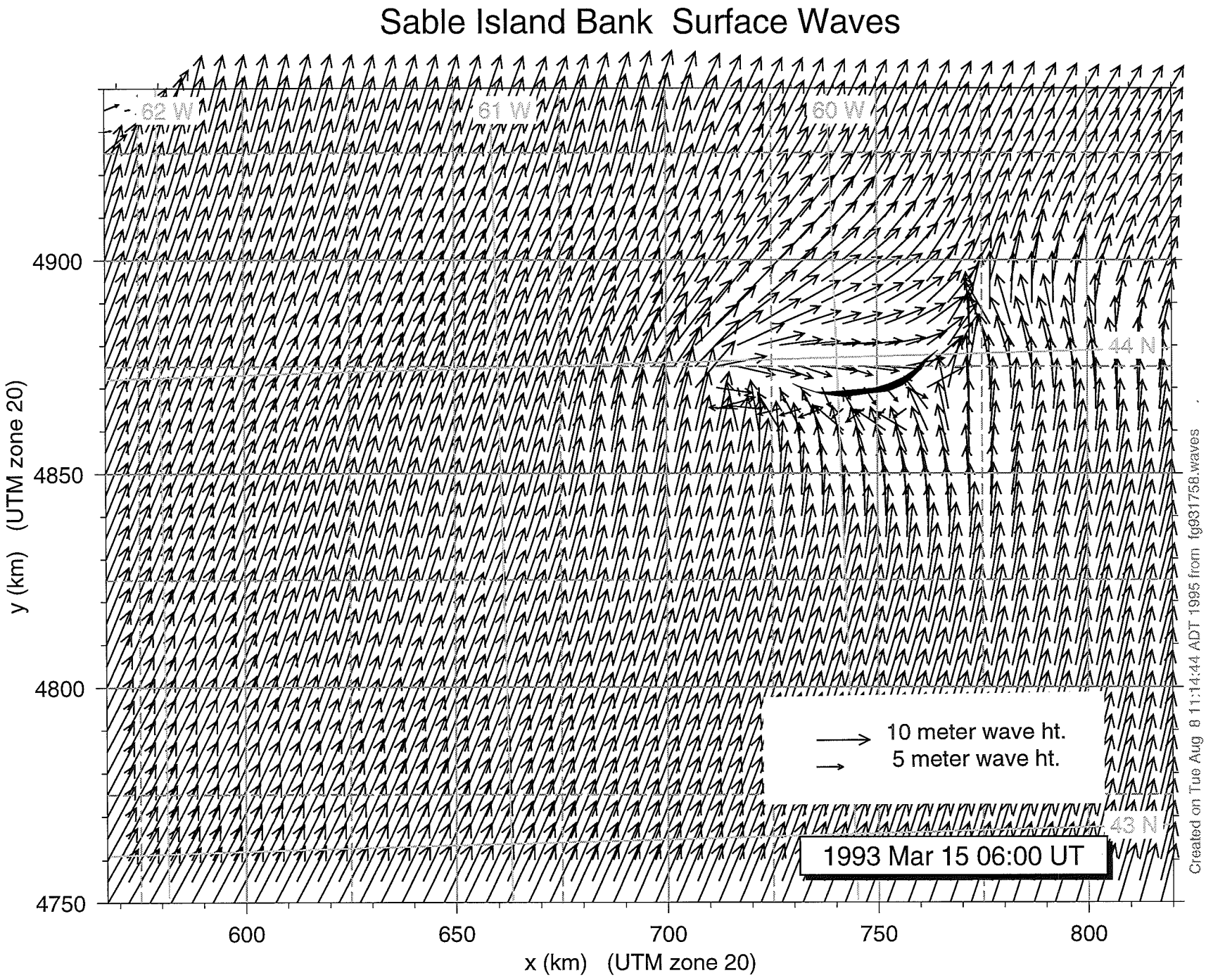

Sable Island Bank Surface Waves
62 W
61 W
60 W
44 N
43 N
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
10 meter wave ht.
5 meter wave ht.
1993 Mar 15 06:00 UT
Created on Tue Aug 8 11:14:44 ADT 1995 from fg931758.waves

Figure 3.5

Sable Island Bank Surface Waves

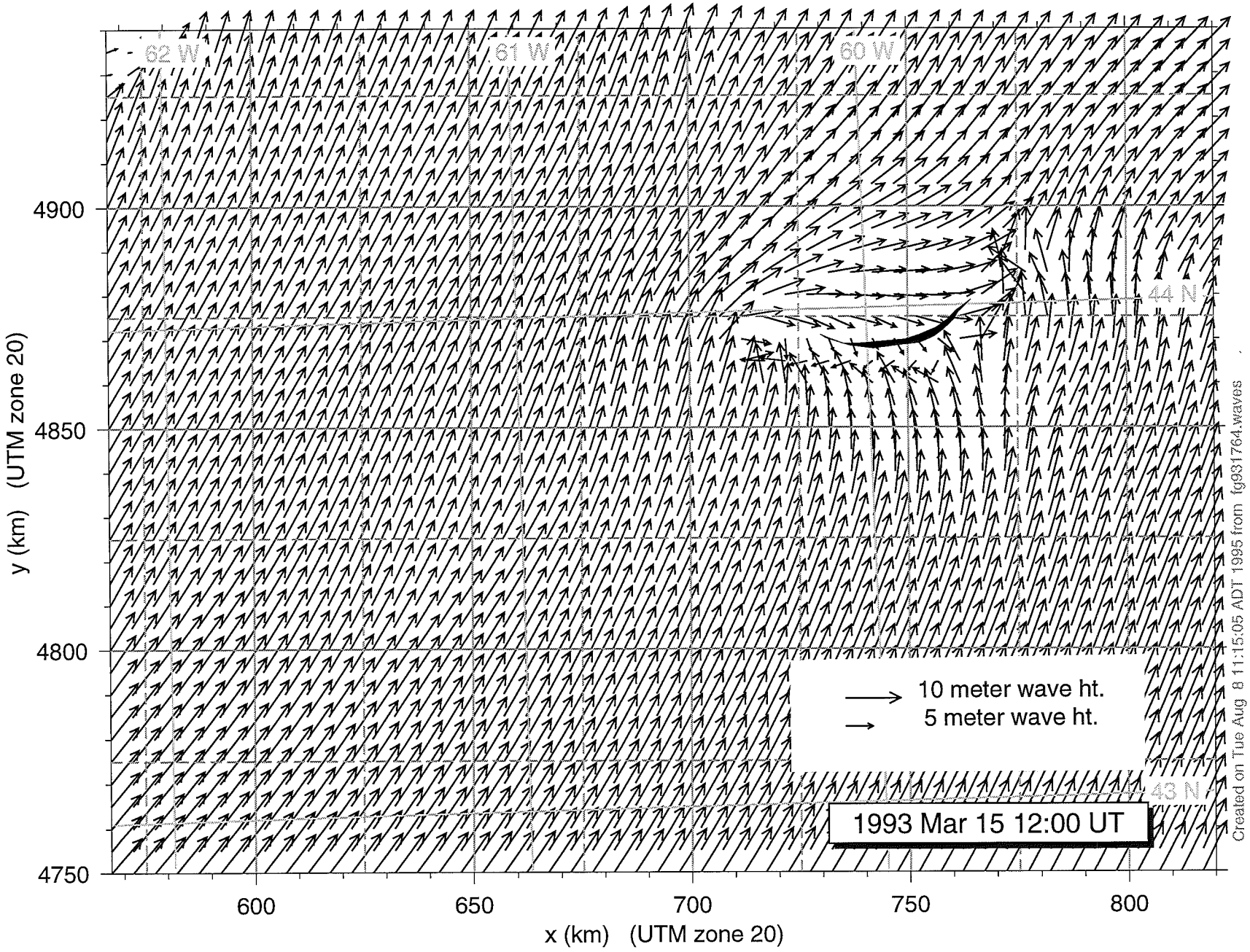


Figure 3.6

Sable Island Bank Surface Waves

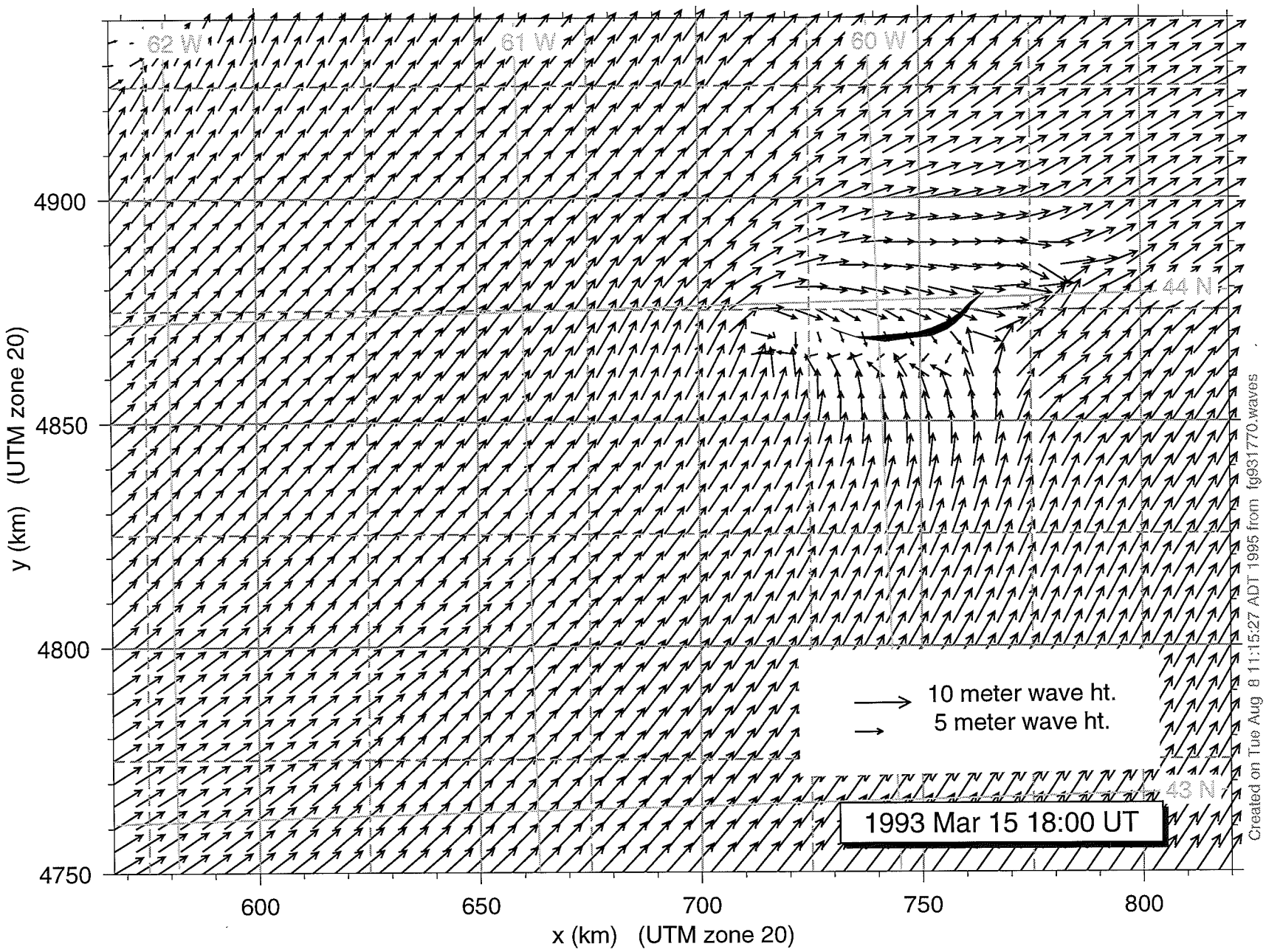


Figure 3.7

Sable Island Bank Surface Waves

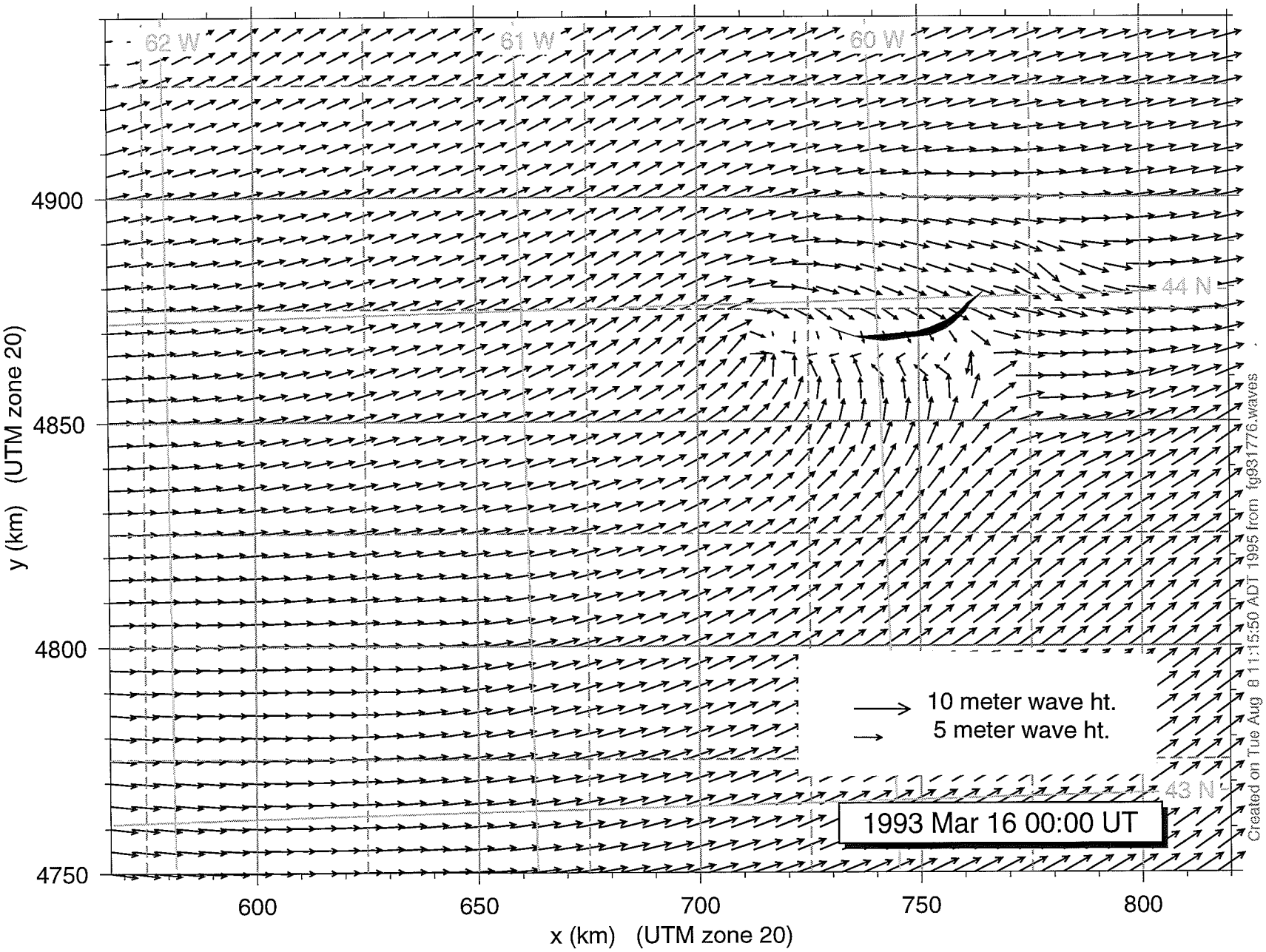


Figure 3.8

Sable Island Bank Surface Waves

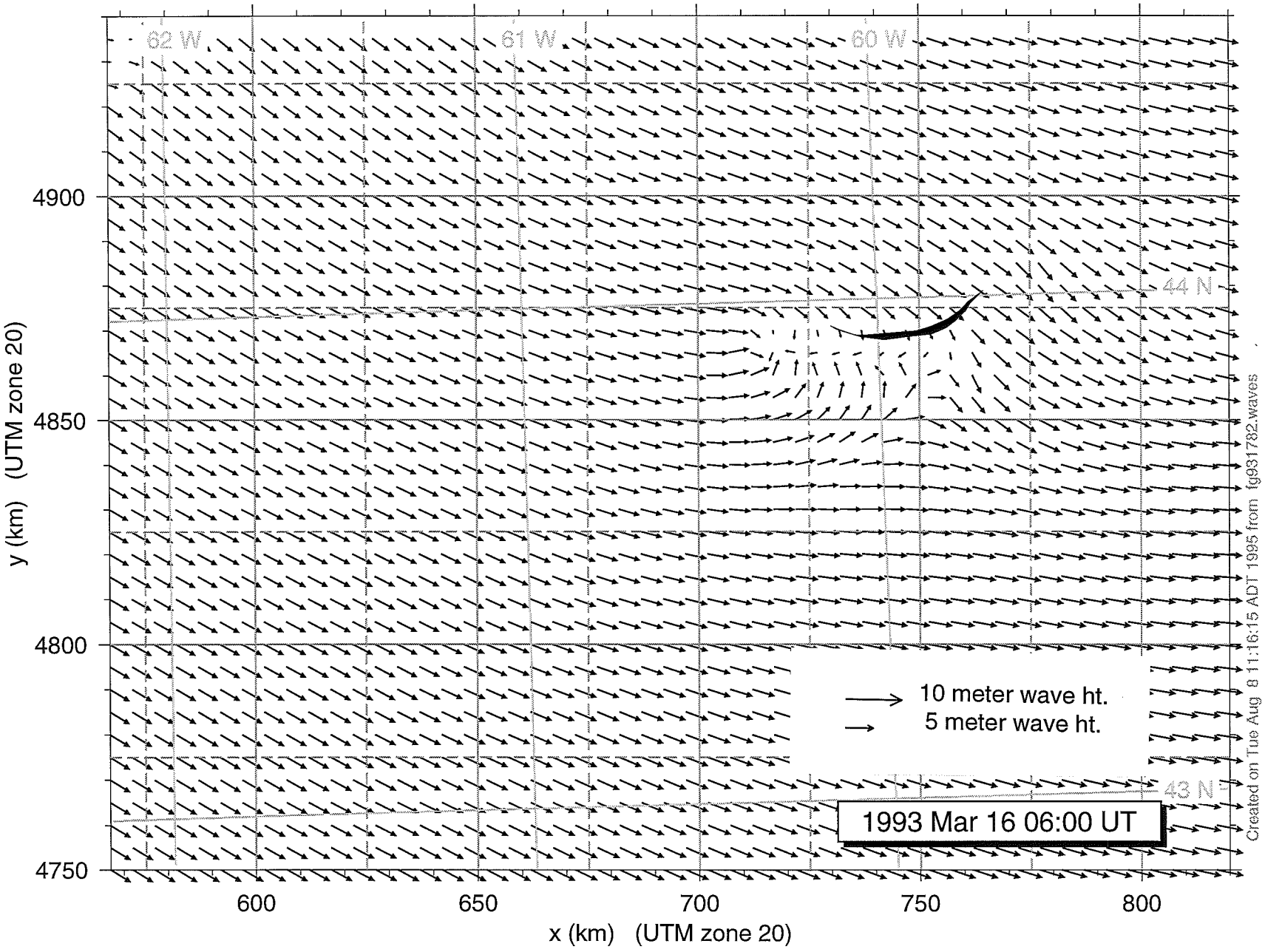


Figure 3.9

Sable Island Bank Surface Waves

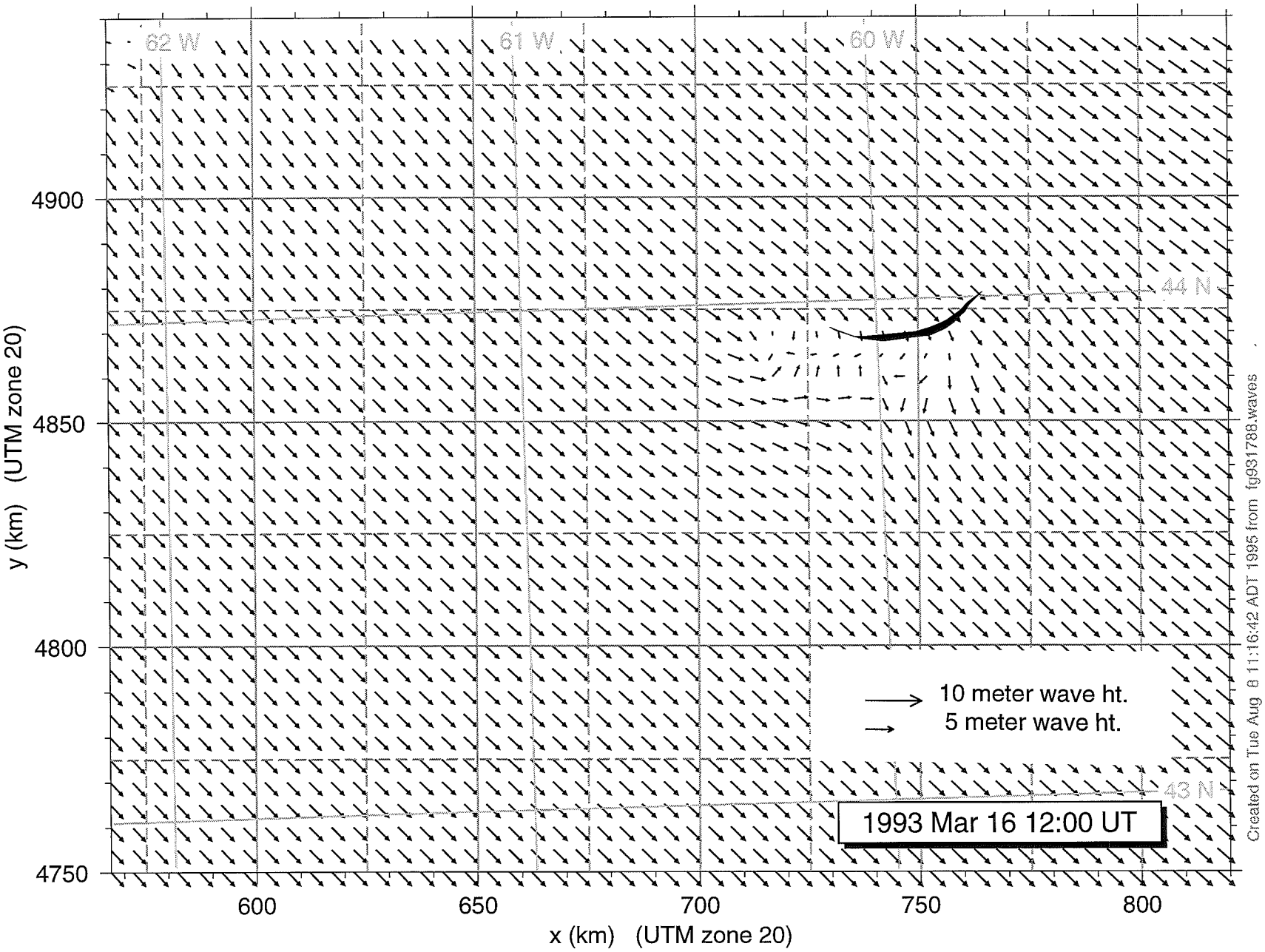


Figure 3.10

Sable Island Bank Surface Waves

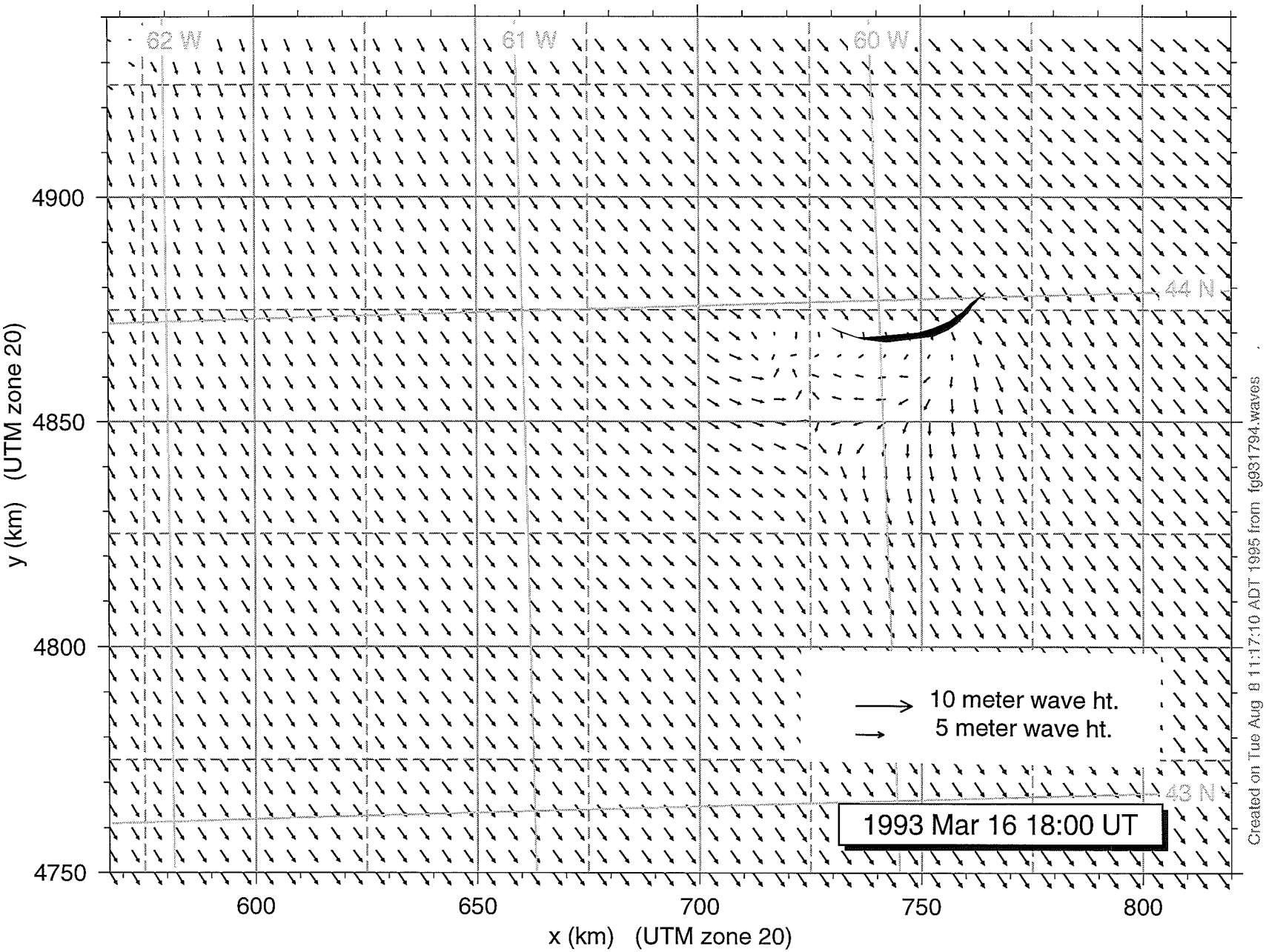


Figure 3.11

Sable Island Bank Surface Waves

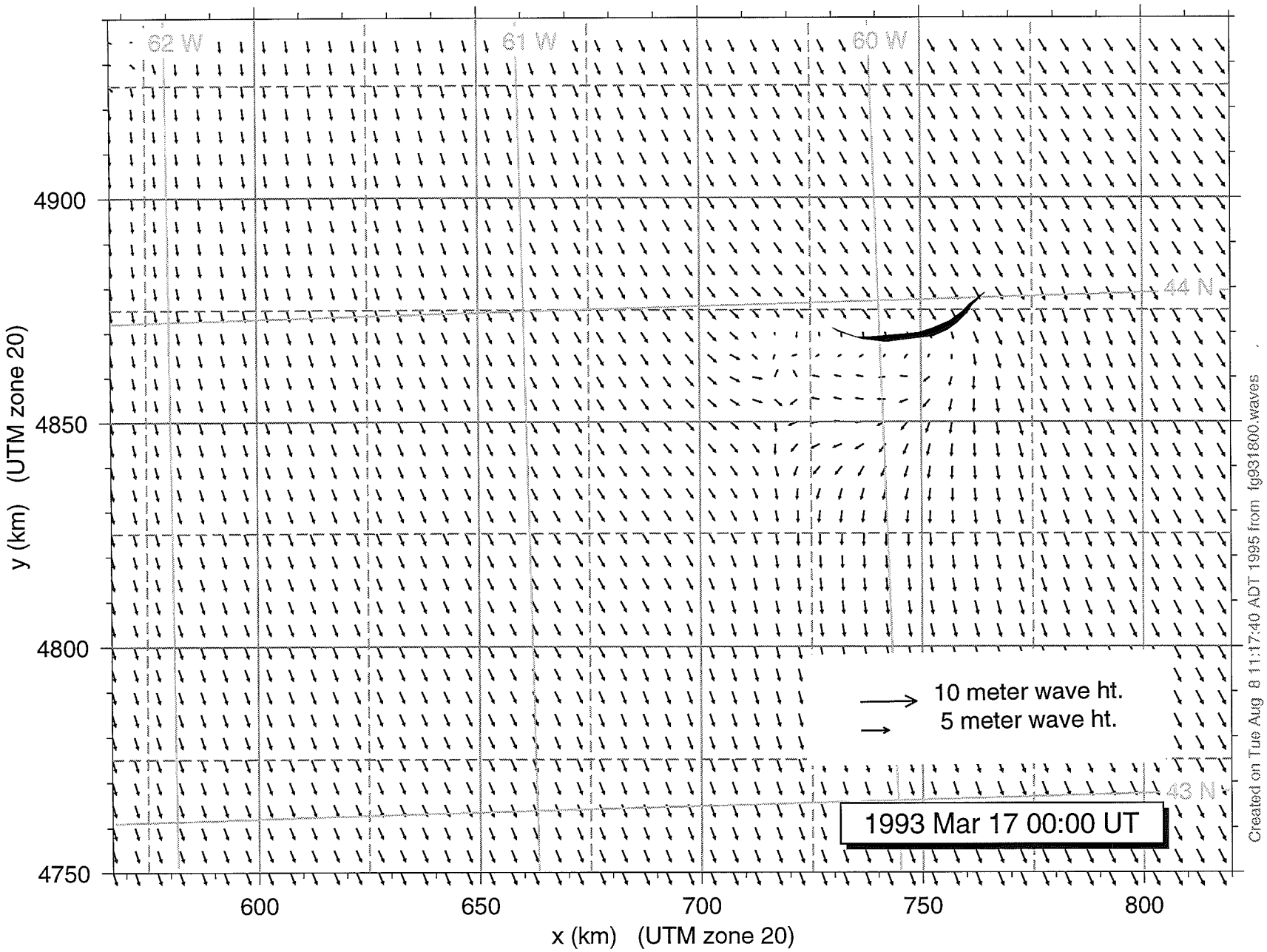


Figure 3.12

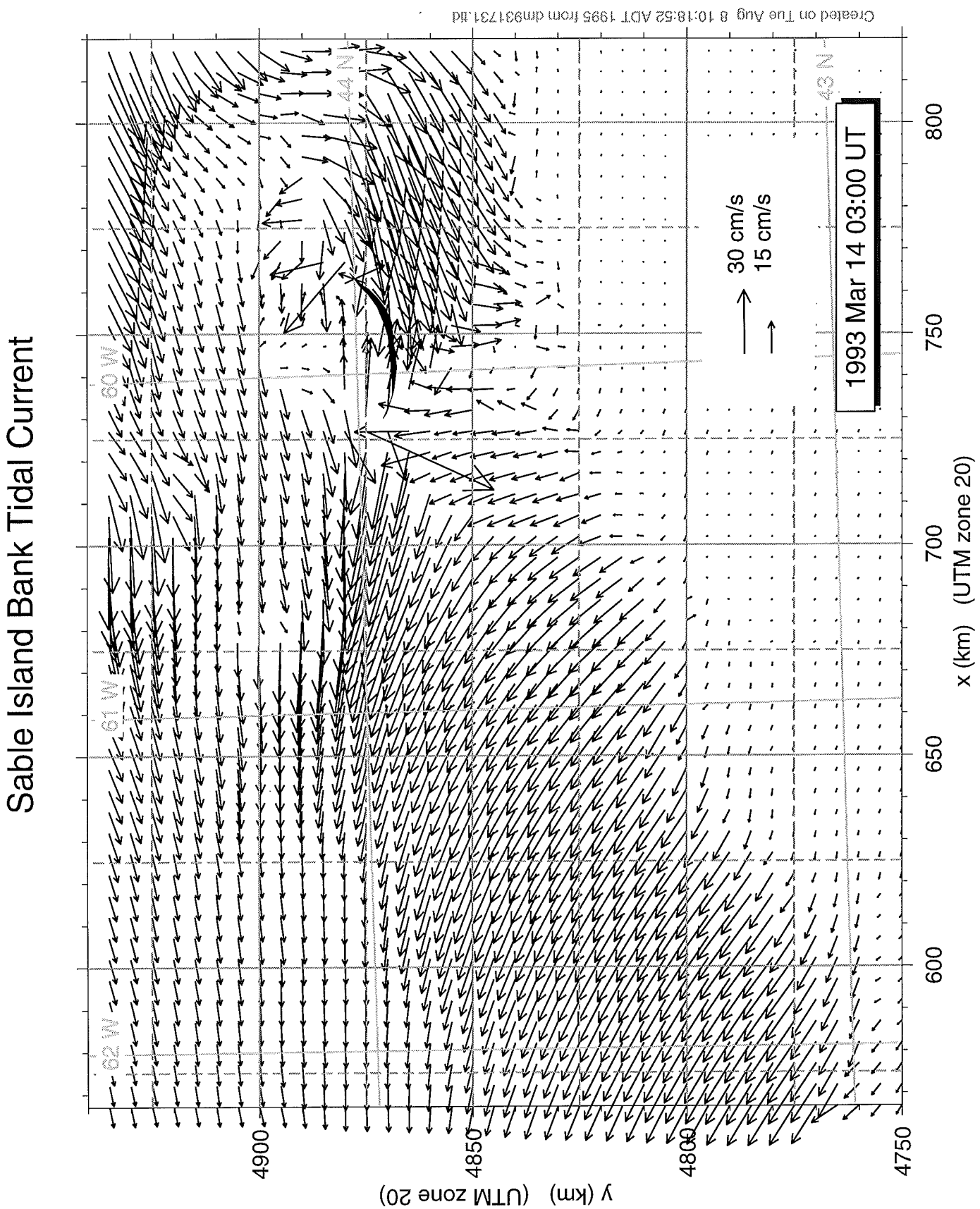

Sable Island Bank Tidal Current
Created on Tue Aug 8 10:18:52 ADT 1995 from dm931731.lid
62 W
61 W
60 W
44 N
43 N
30 cm/s
15 cm/s
1993 Mar 14 03:00 UT
600
650
700
750
800
x (km) (UTM zone 20)
4900
4850
4800
4750
y (km) (UTM zone 20)

Figure 4.1

Sable Island Bank Tidal Current

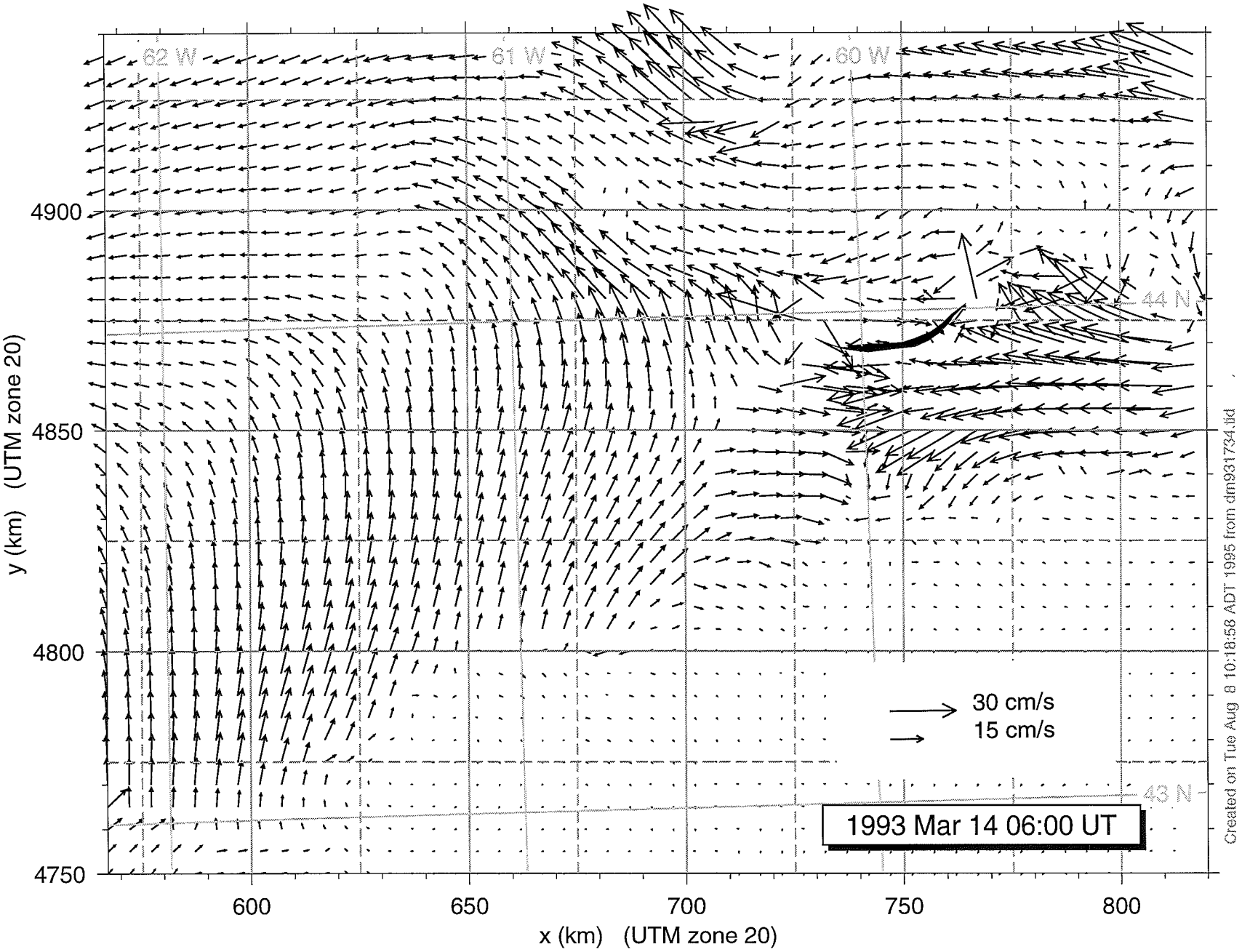


Figure 4.2

Sable Island Bank Tidal Current

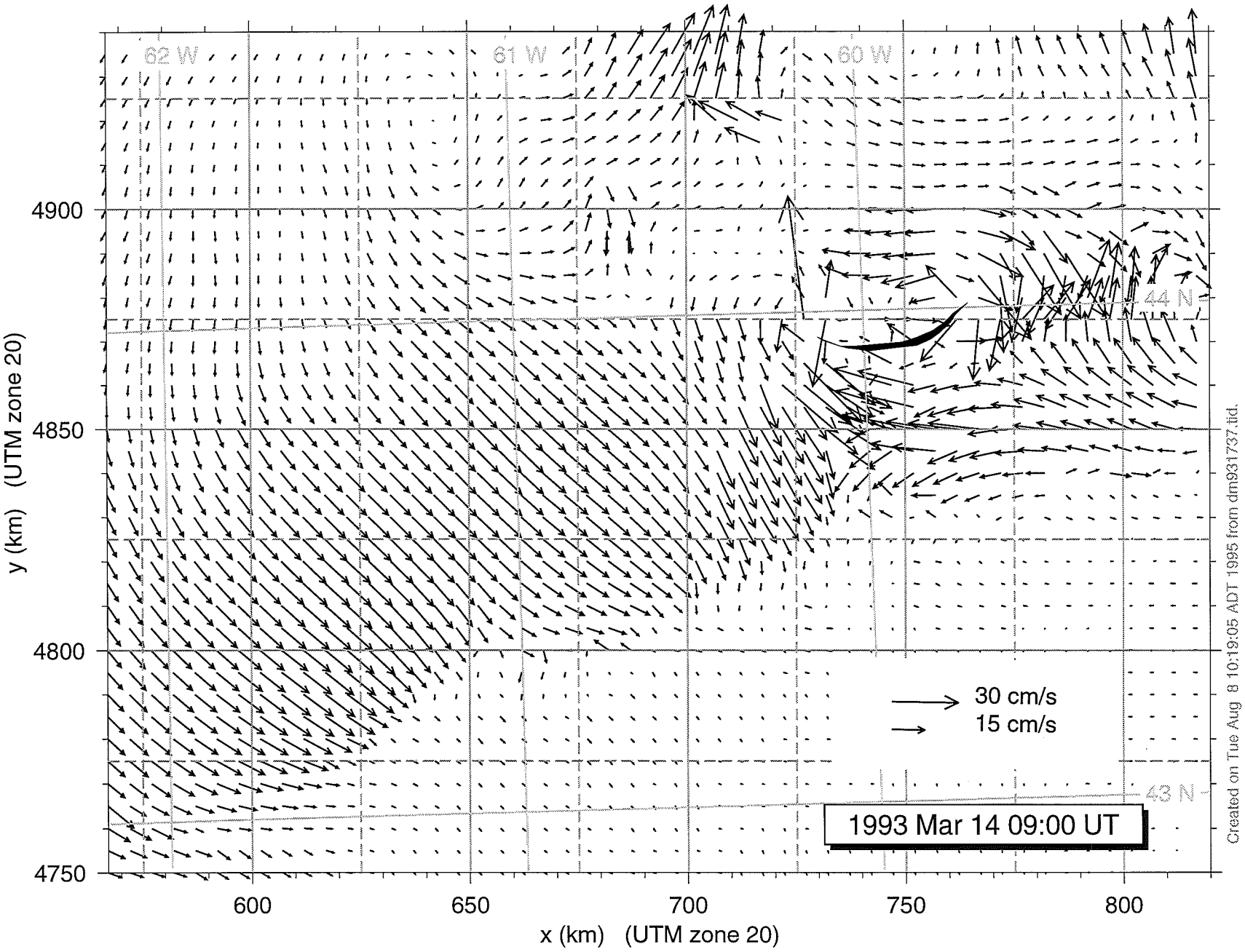


Figure 4.3

Sable Island Bank Tidal Current

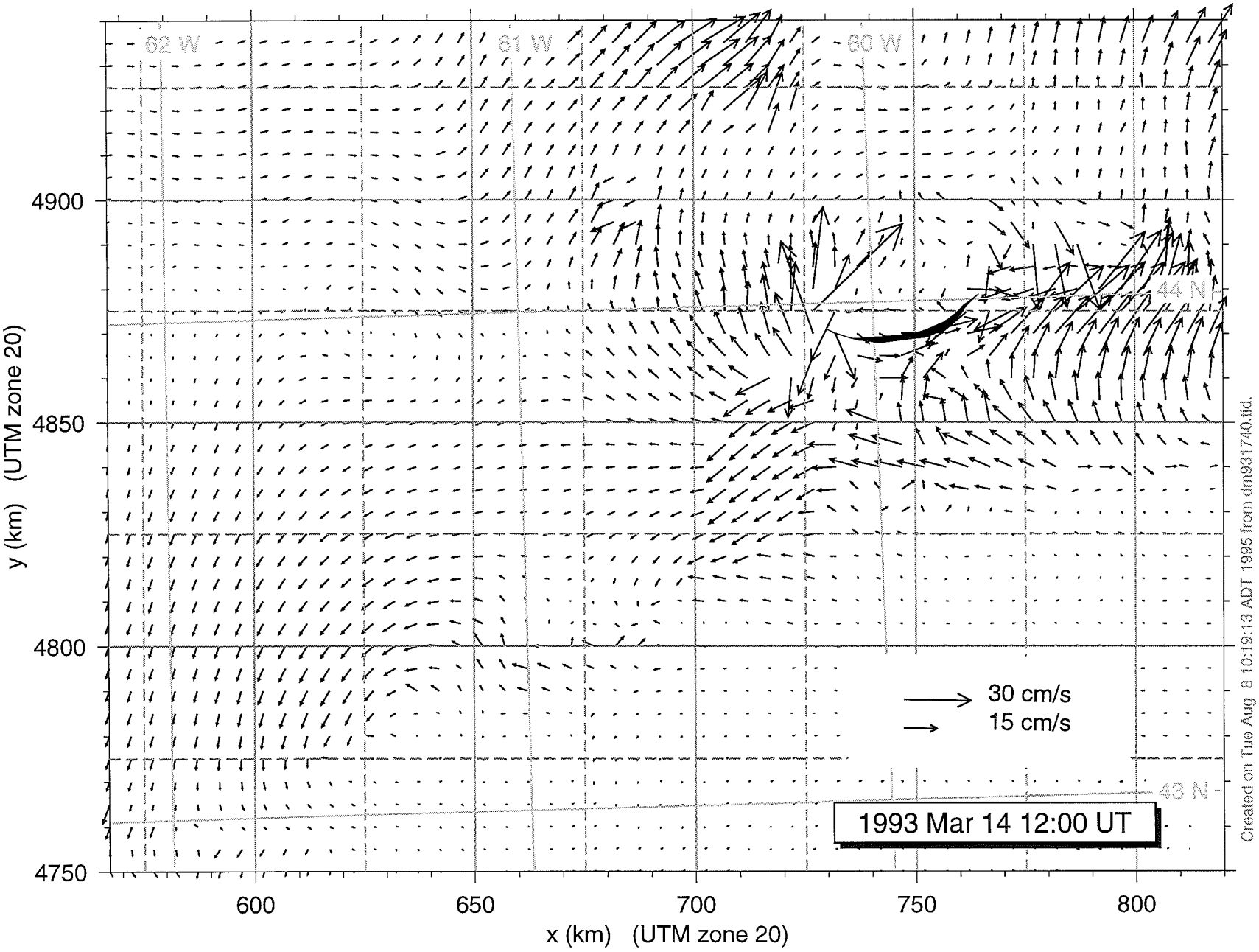


Figure 4.4

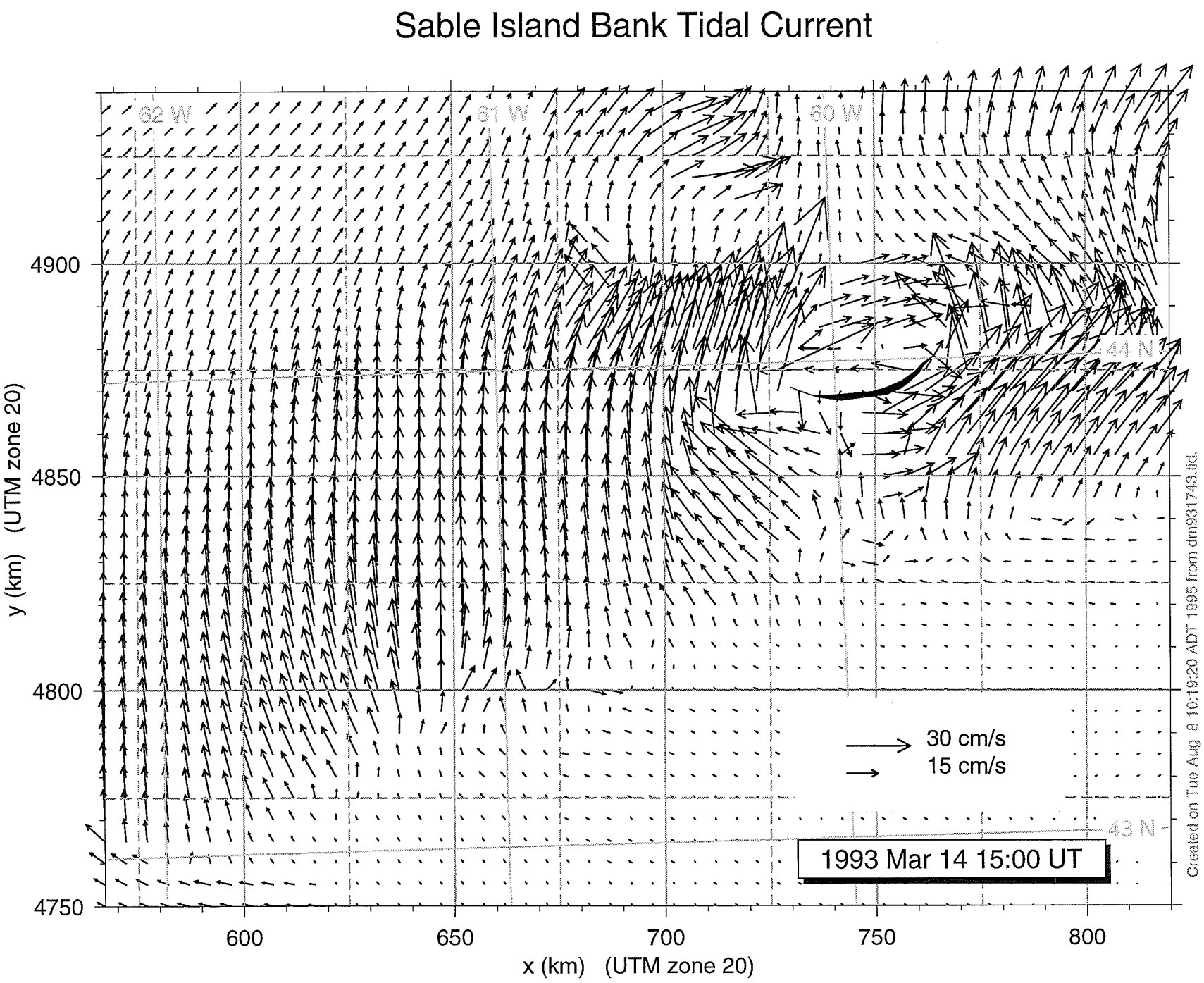

Sable Island Bank Tidal Current
62 W
61 W
60 W
44 N
43 N
30 cm/s
15 cm/s
1993 Mar 14 15:00 UT
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
Created on Tue Aug 8 10:19:20 ADT 1995 from dm931743.tid.

Figure 4.5

Sable Island Bank Tidal Current

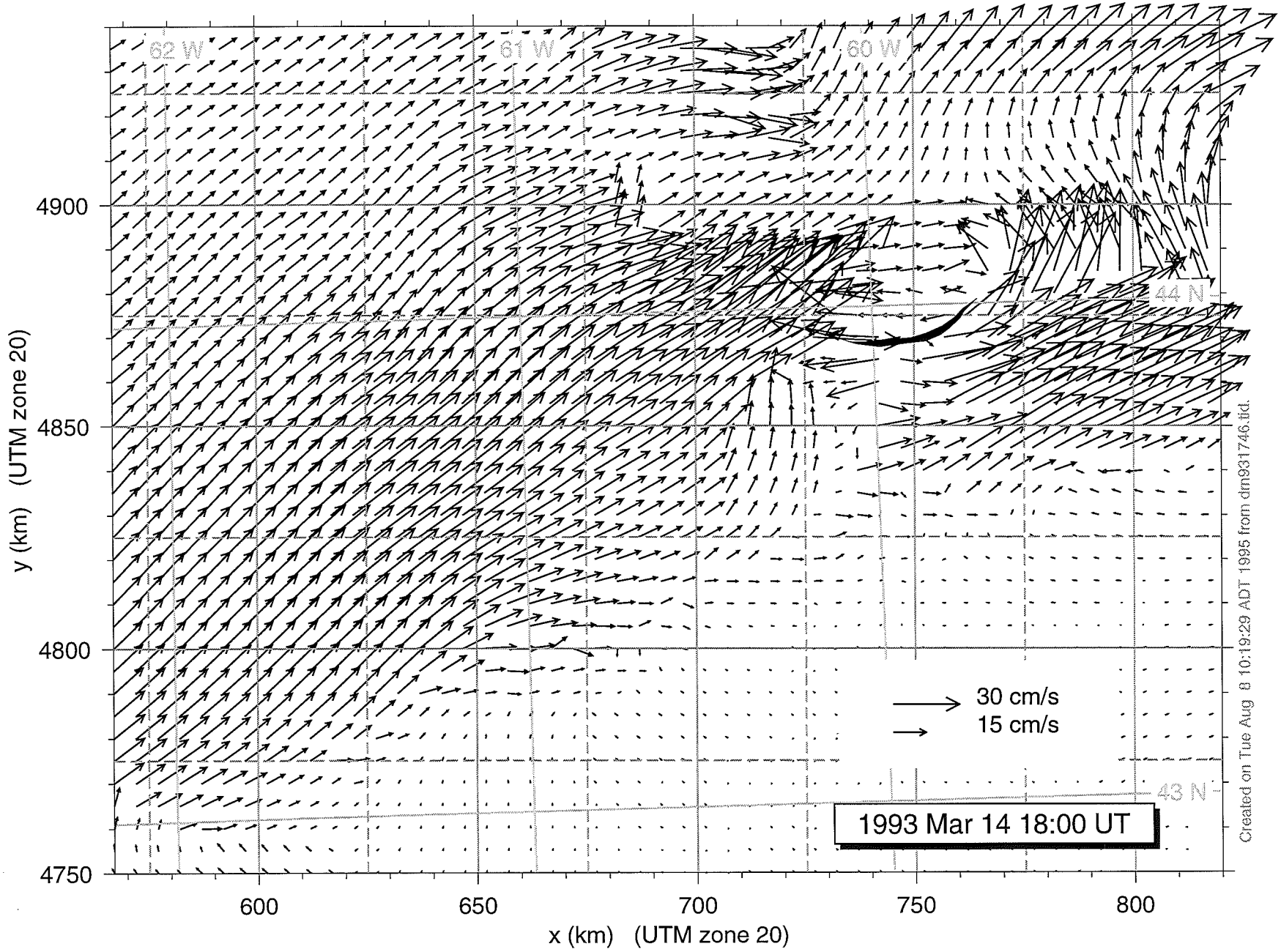


Figure 4.6

Sable Island Bank Tidal Current

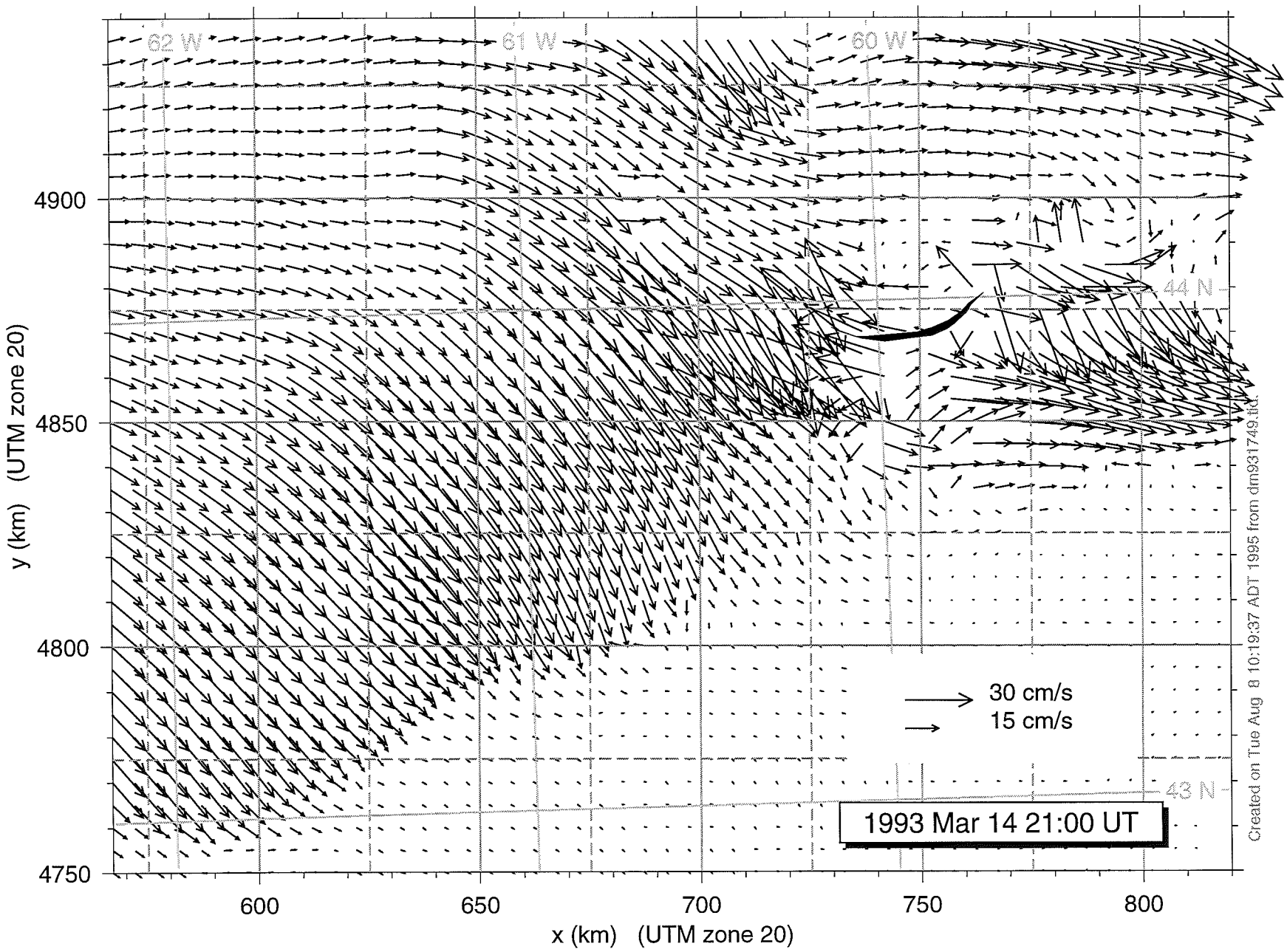


Figure 4.7

Sable Island Bank Tidal Current

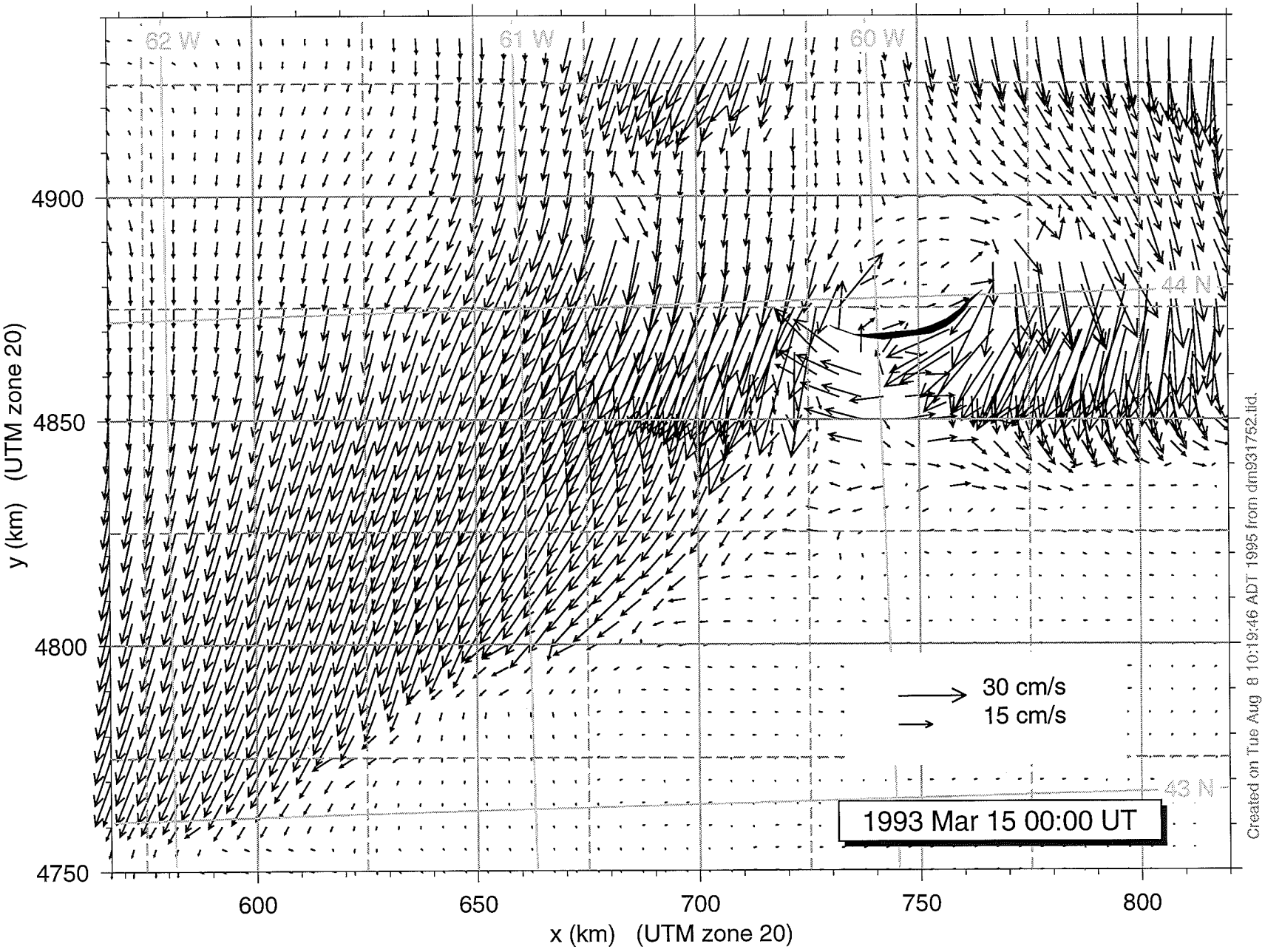


Figure 4.8

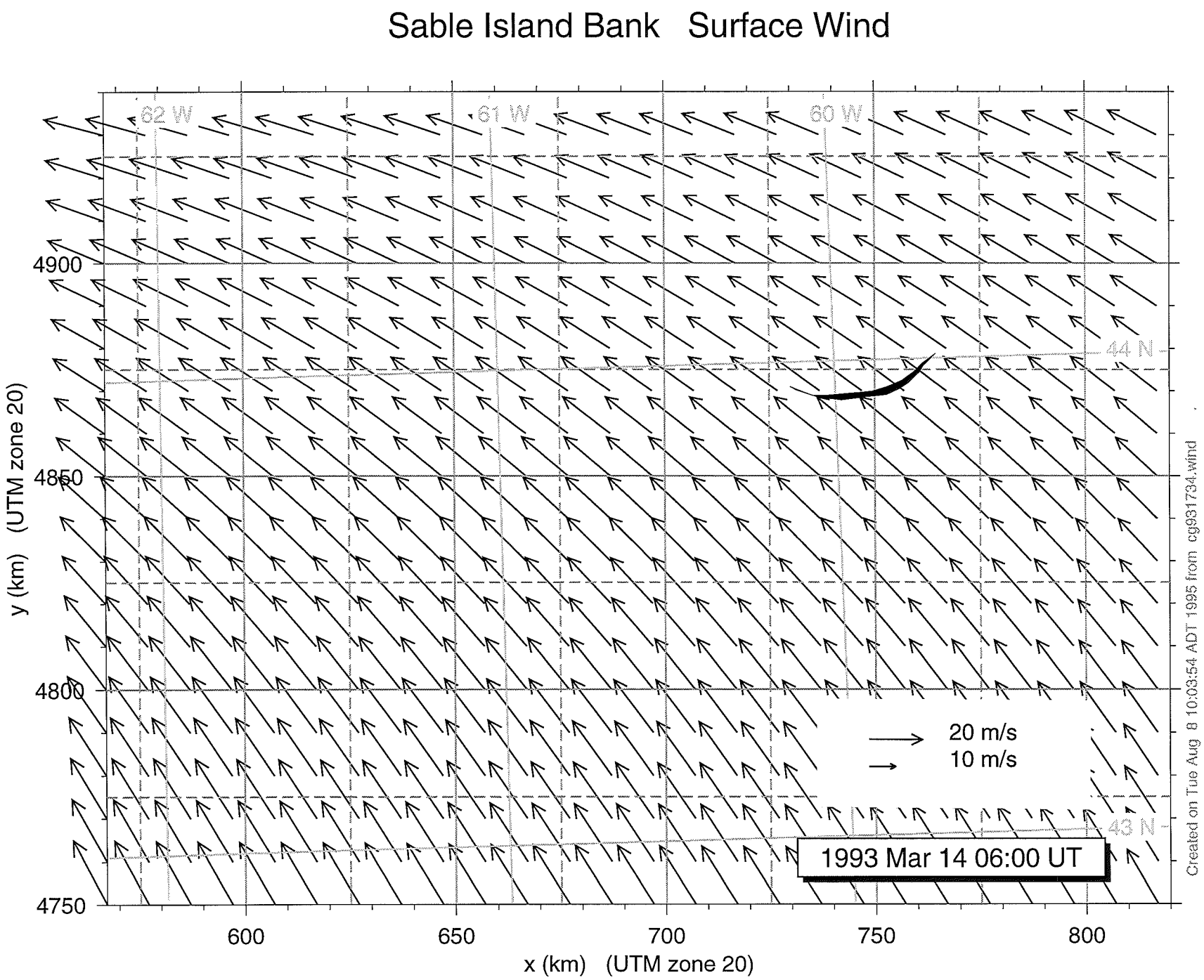

Sable Island Bank Surface Wind
62 W
61 W
60 W
44 N
43 N
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
20 m/s
10 m/s
1993 Mar 14 06:00 UT
Created on Tue Aug 8 10:03:54 ADT 1995 from cg931734.wind

Figure 5.1

Sable Island Bank Tidal Current

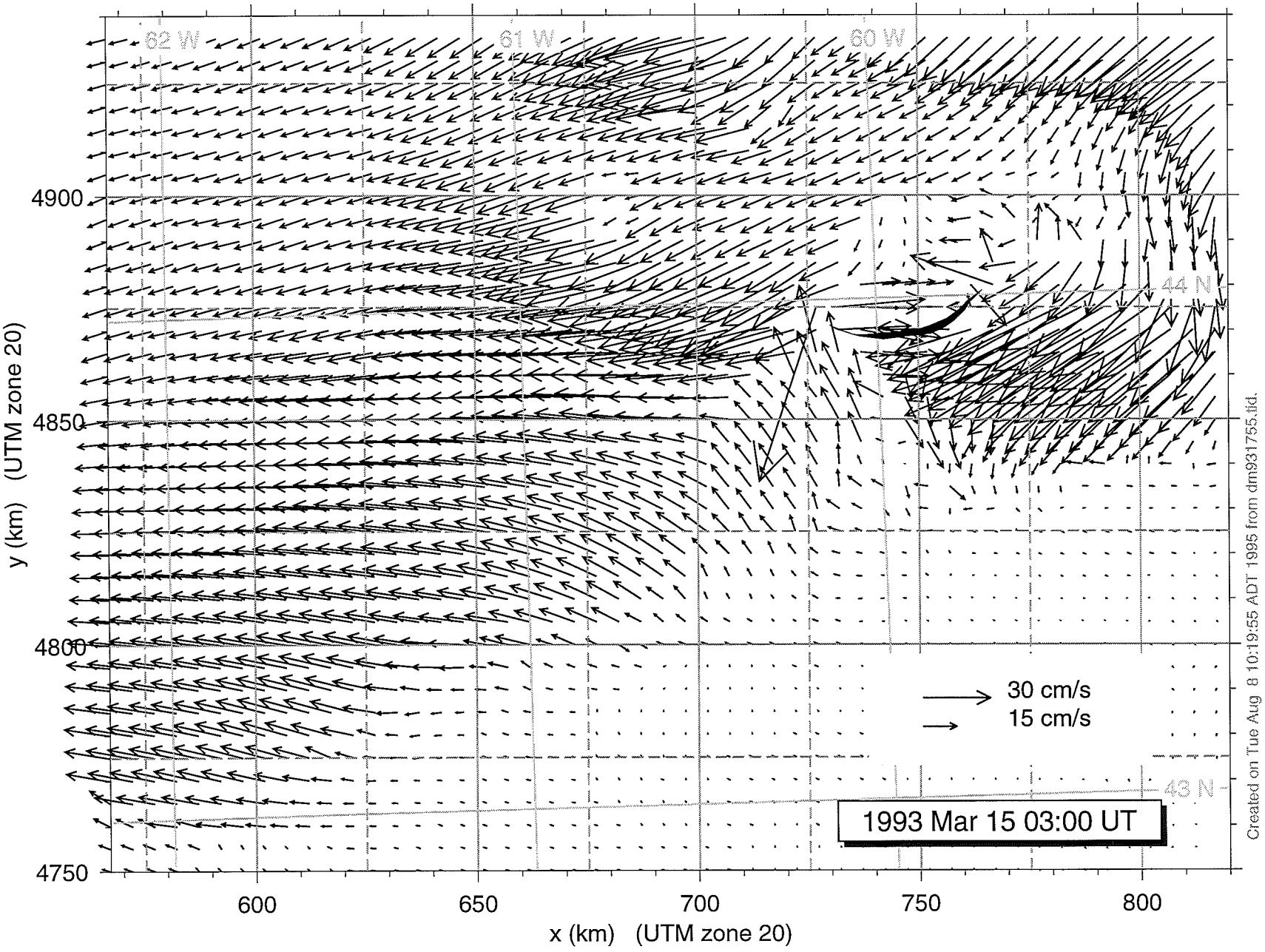


Figure 4.9

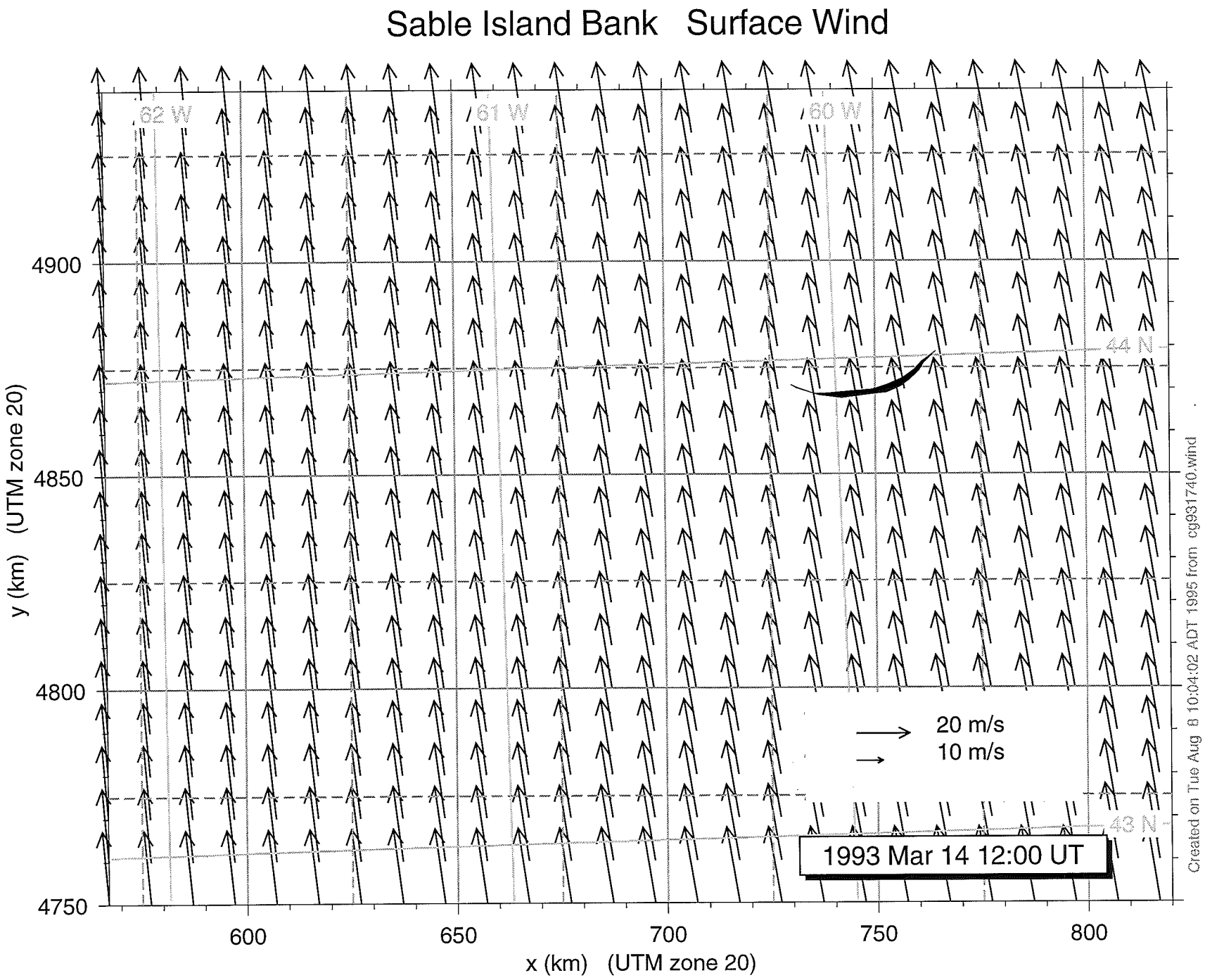

Sable Island Bank Surface Wind
62 W
61 W
60 W
44 N
43 N
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
20 m/s
10 m/s
1993 Mar 14 12:00 UT
Created on Tue Aug 8 10:04:02 ADT 1995 from cg931740.wind

Figure 5.2

Sable Island Bank Surface Wind

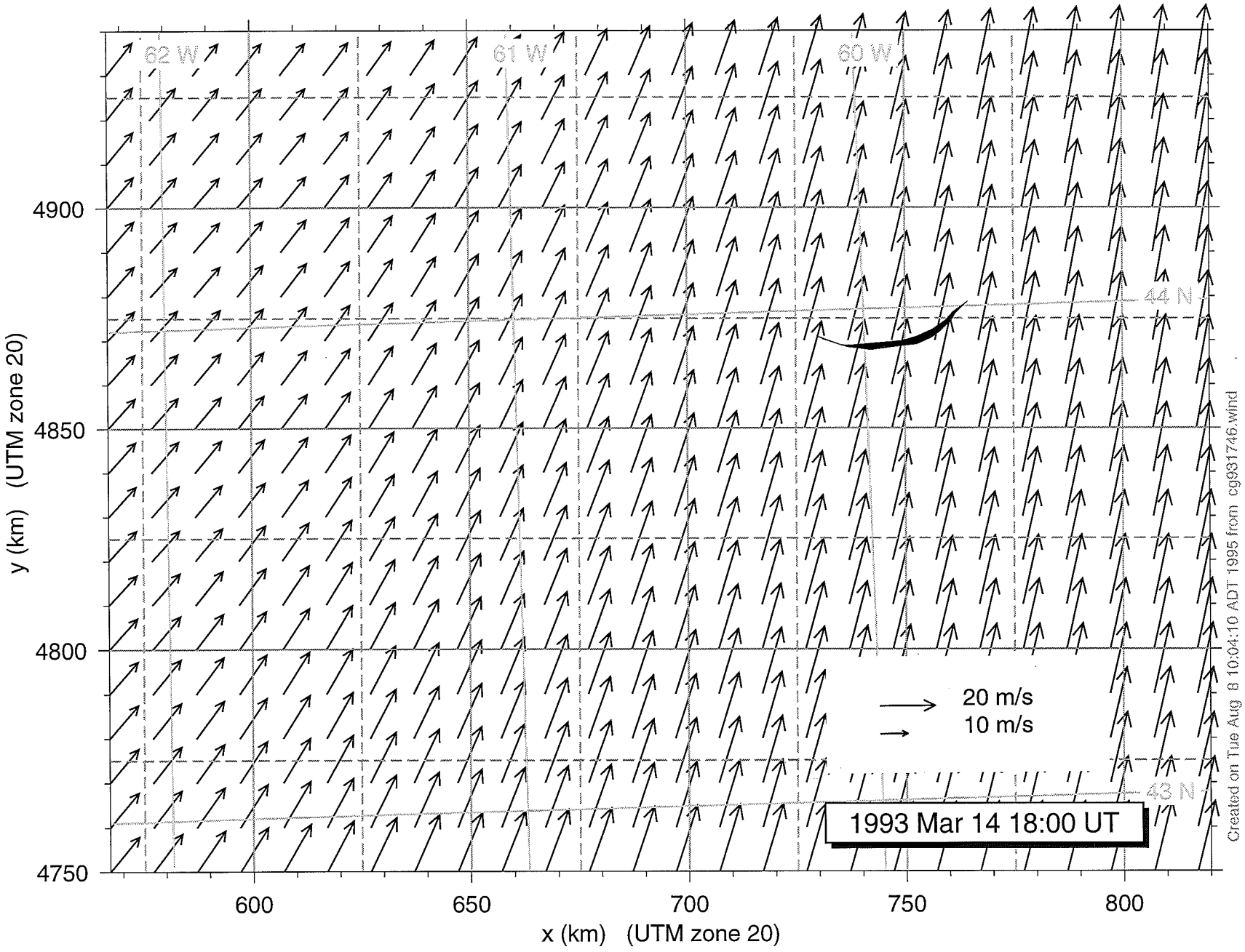


Figure 5.3

Sable Island Bank Surface Wind

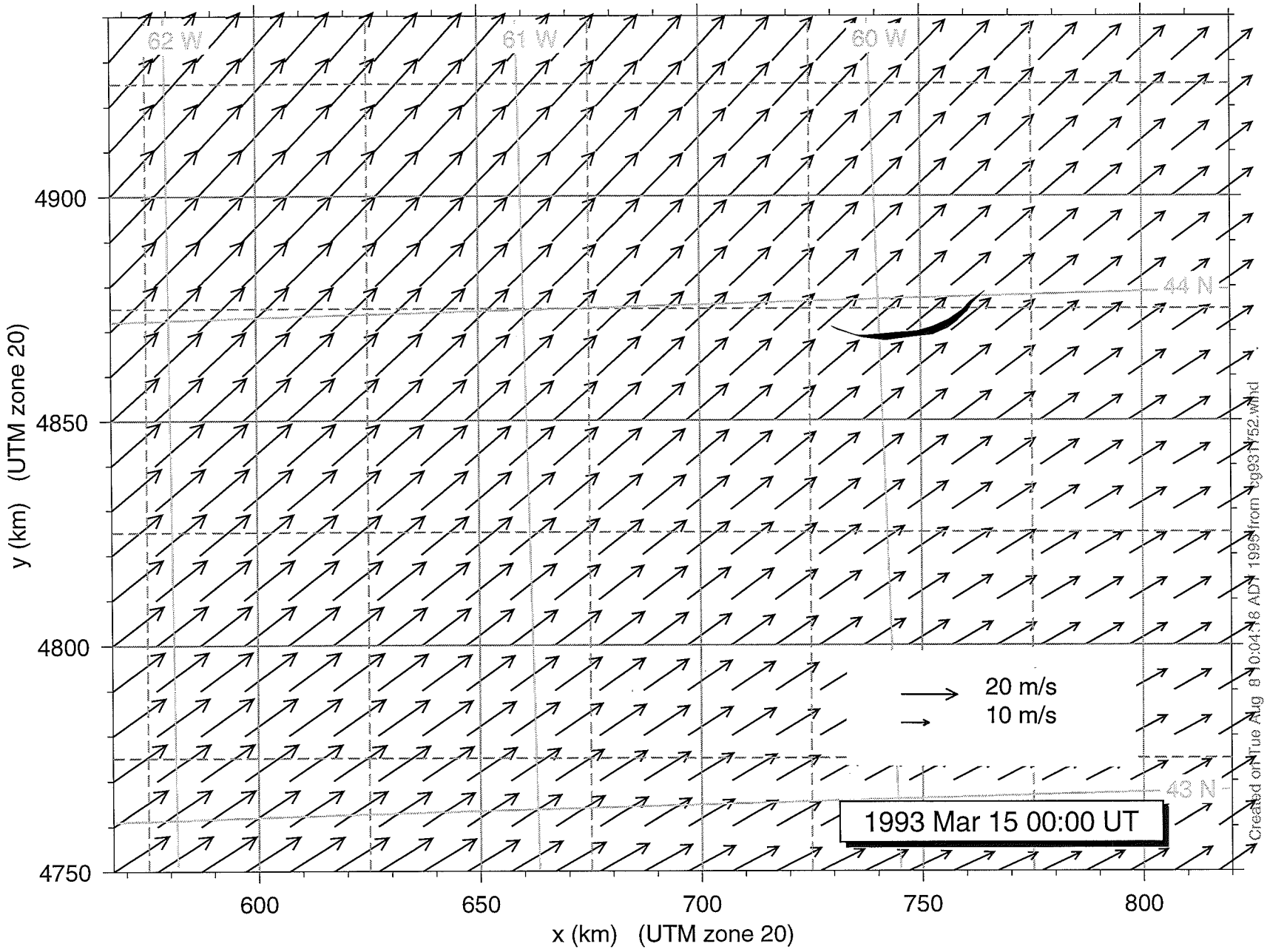


Figure 5.4

Sable Island Bank Surface Wind

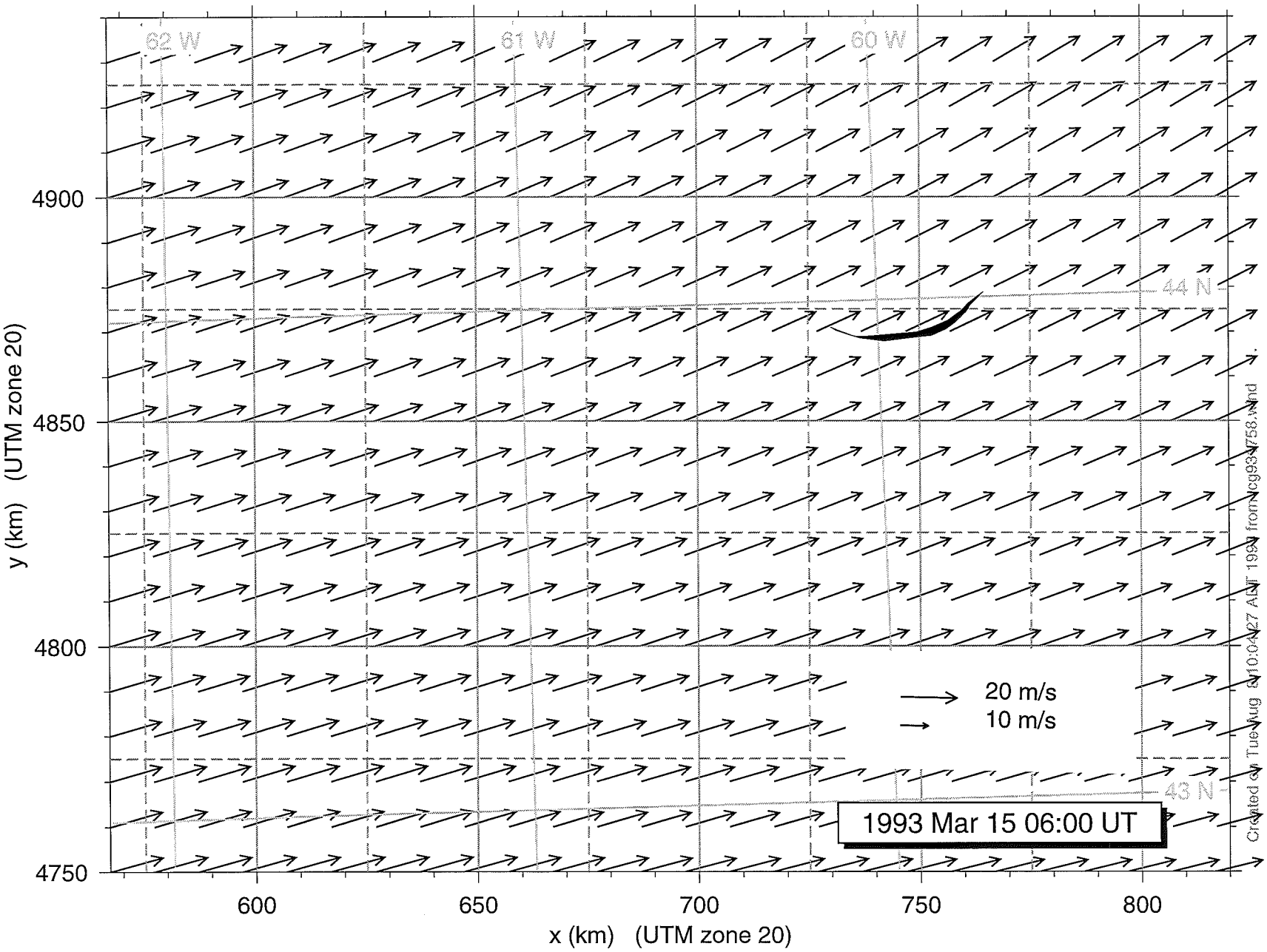


Figure 5.5

Sable Island Bank Surface Wind

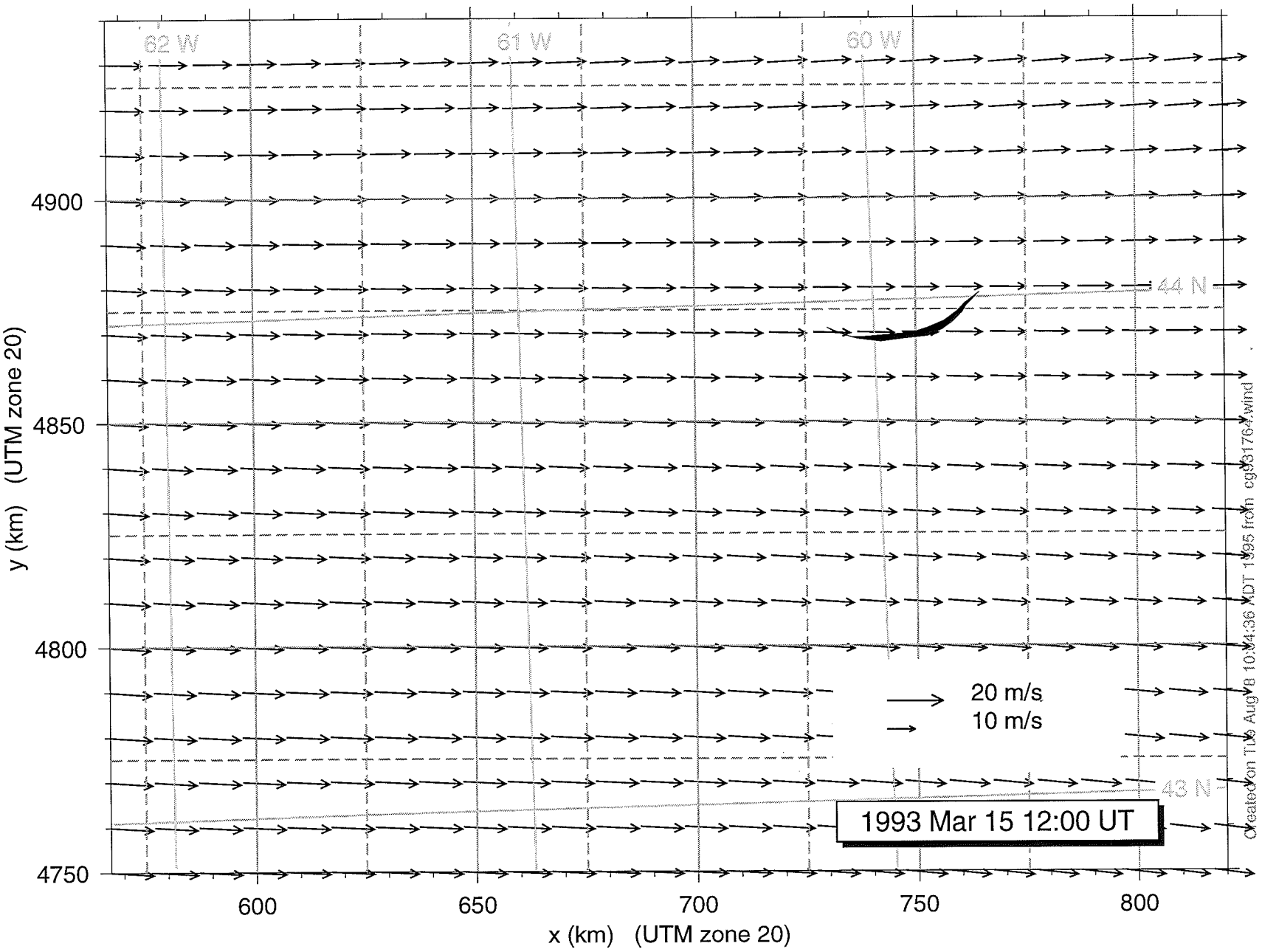


Figure 5.6

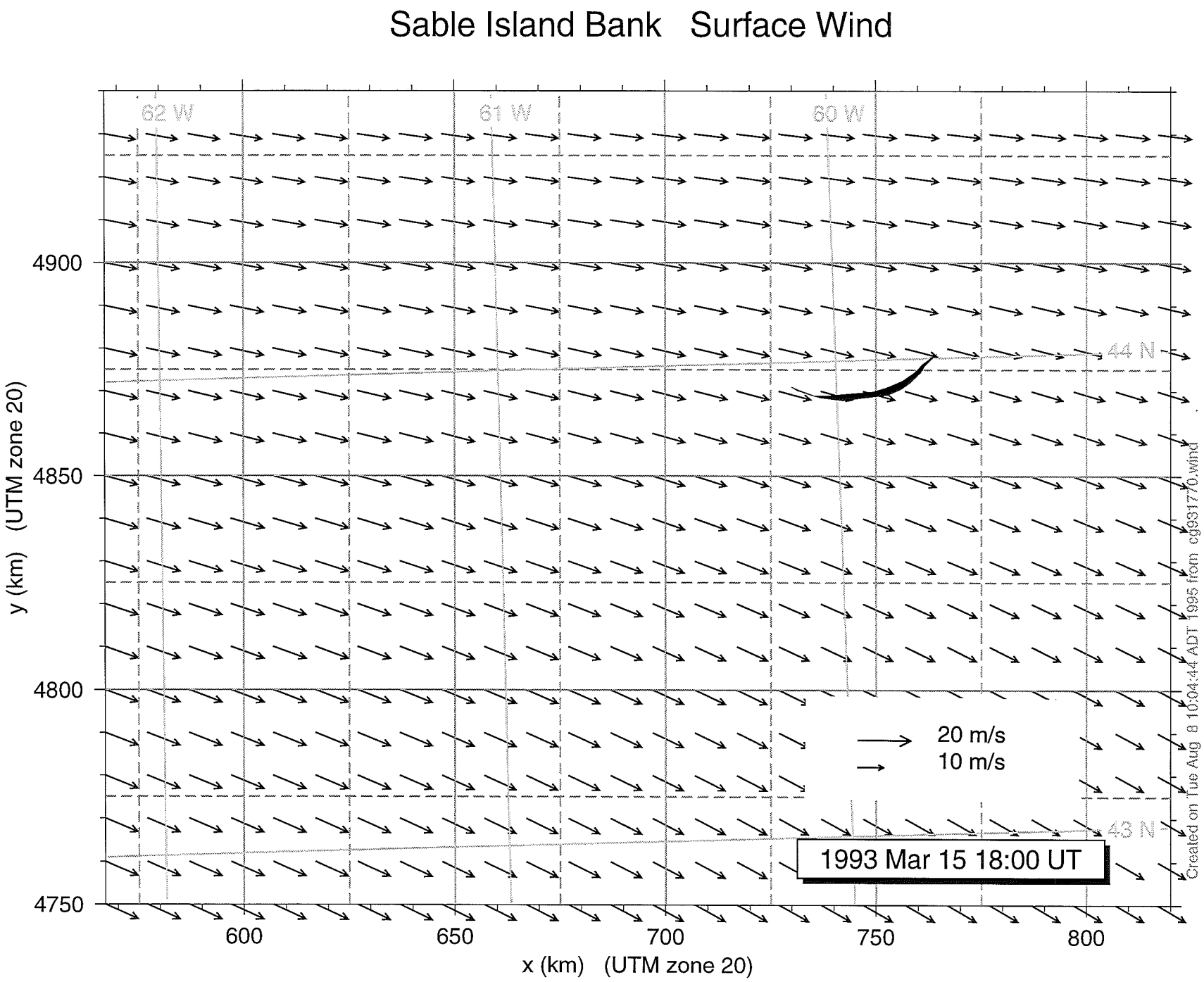

Sable Island Bank Surface Wind
62 W
61 W
60 W
44 N
43 N
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
20 m/s
10 m/s
1993 Mar 15 18:00 UT
Created on Tue Aug 8 10:04:44 ADT 1995 from cg931770.wind

Figure 5.7

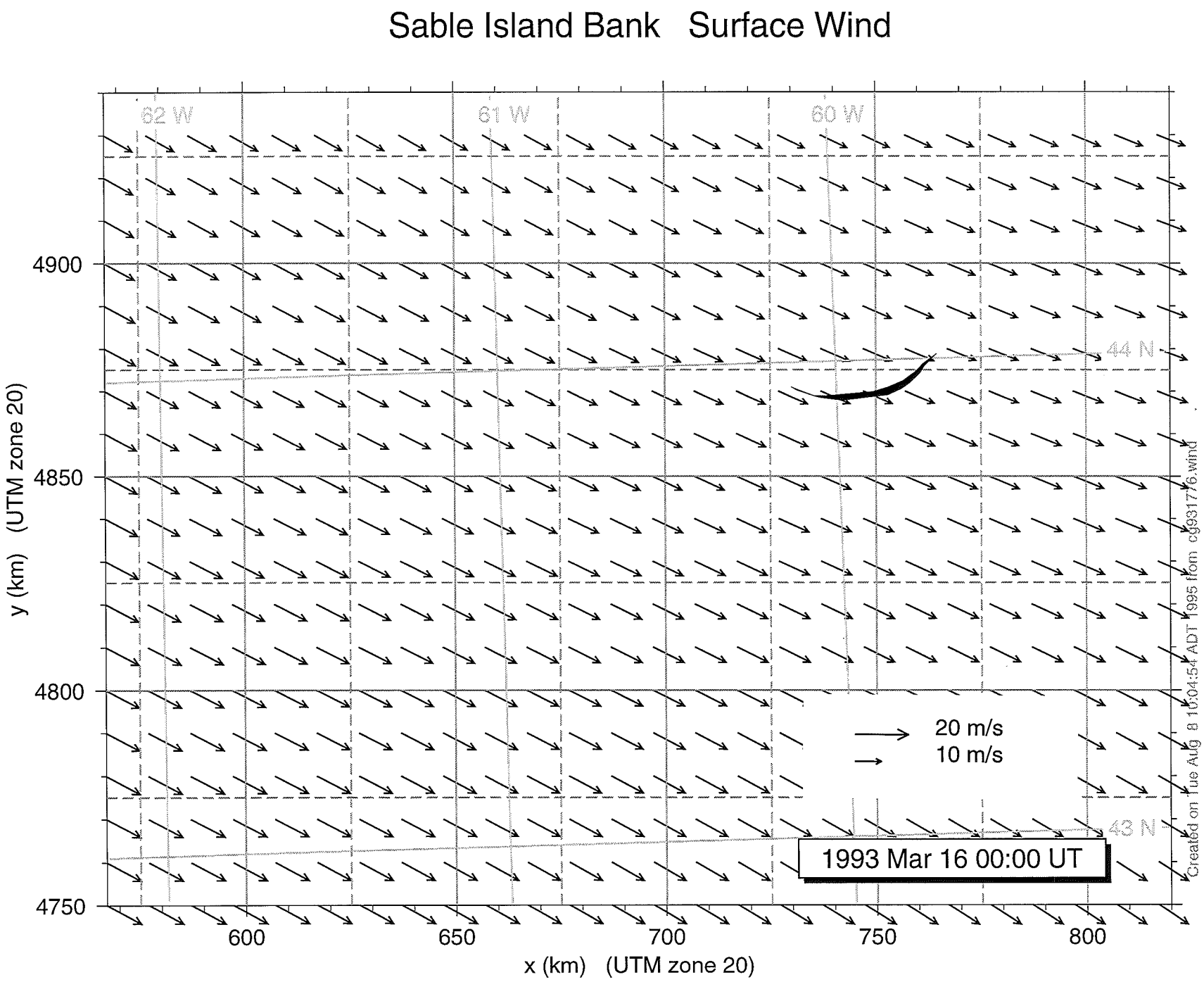

Sable Island Bank Surface Wind
62 W
61 W
60 W
44 N
43 N
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
20 m/s
10 m/s
1993 Mar 16 00:00 UT
Created on Tue Aug 8 10:04:54 ADT 1995 from cg931776.wind

Figure 5.8

Sable Island Bank Surface Wind

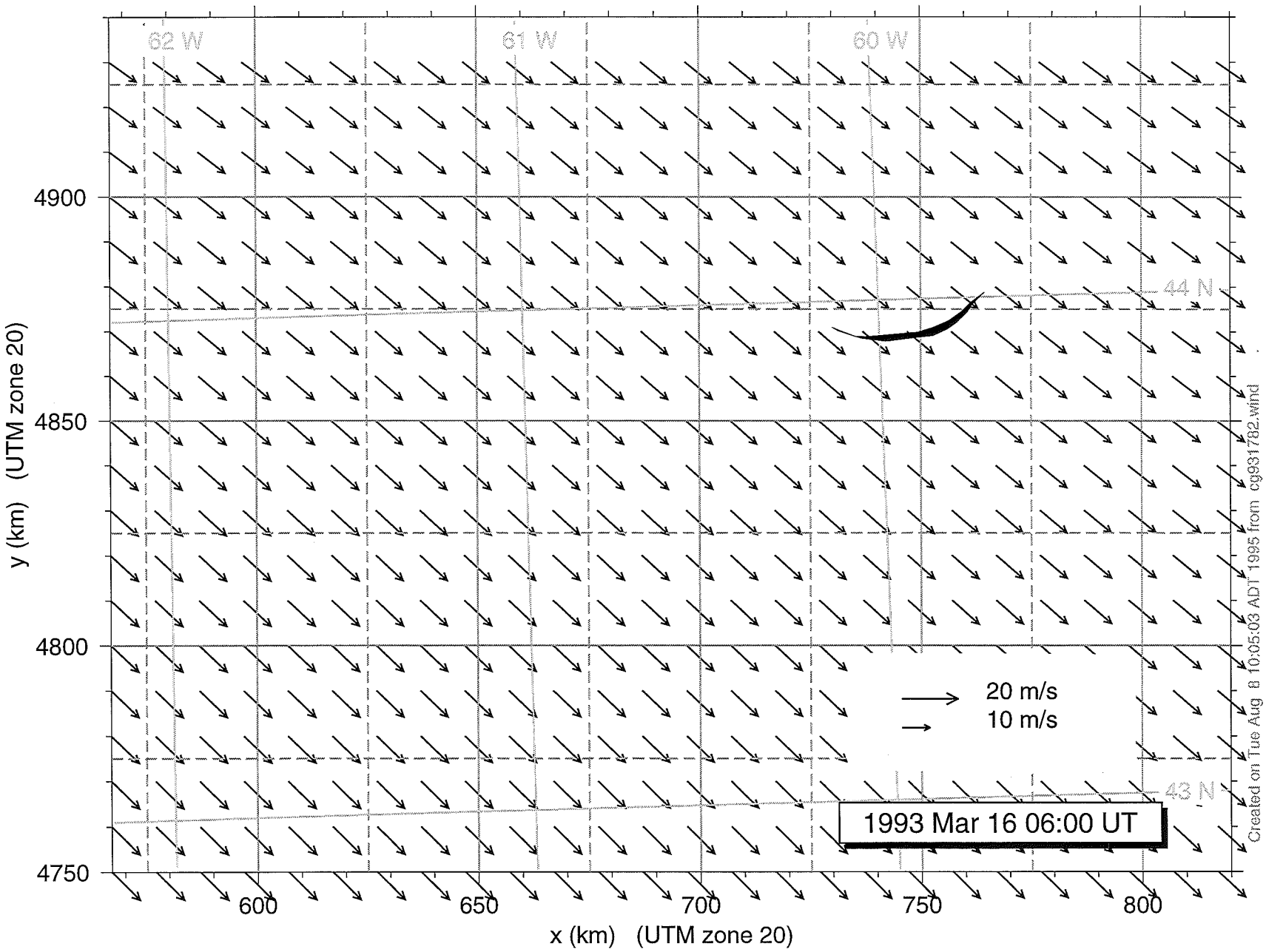


Figure 5.9

Sable Island Bank Surface Wind

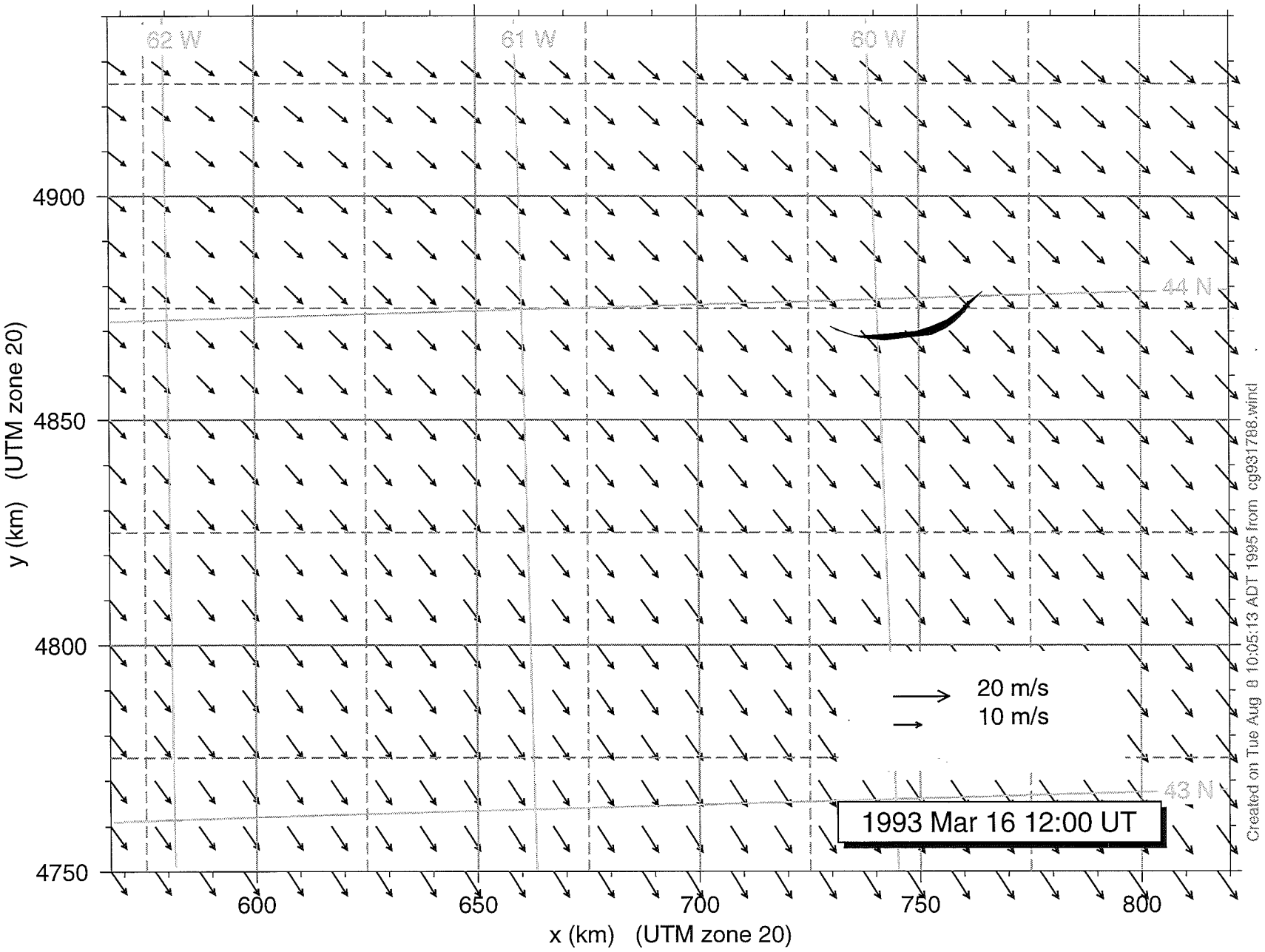


Figure 5.10

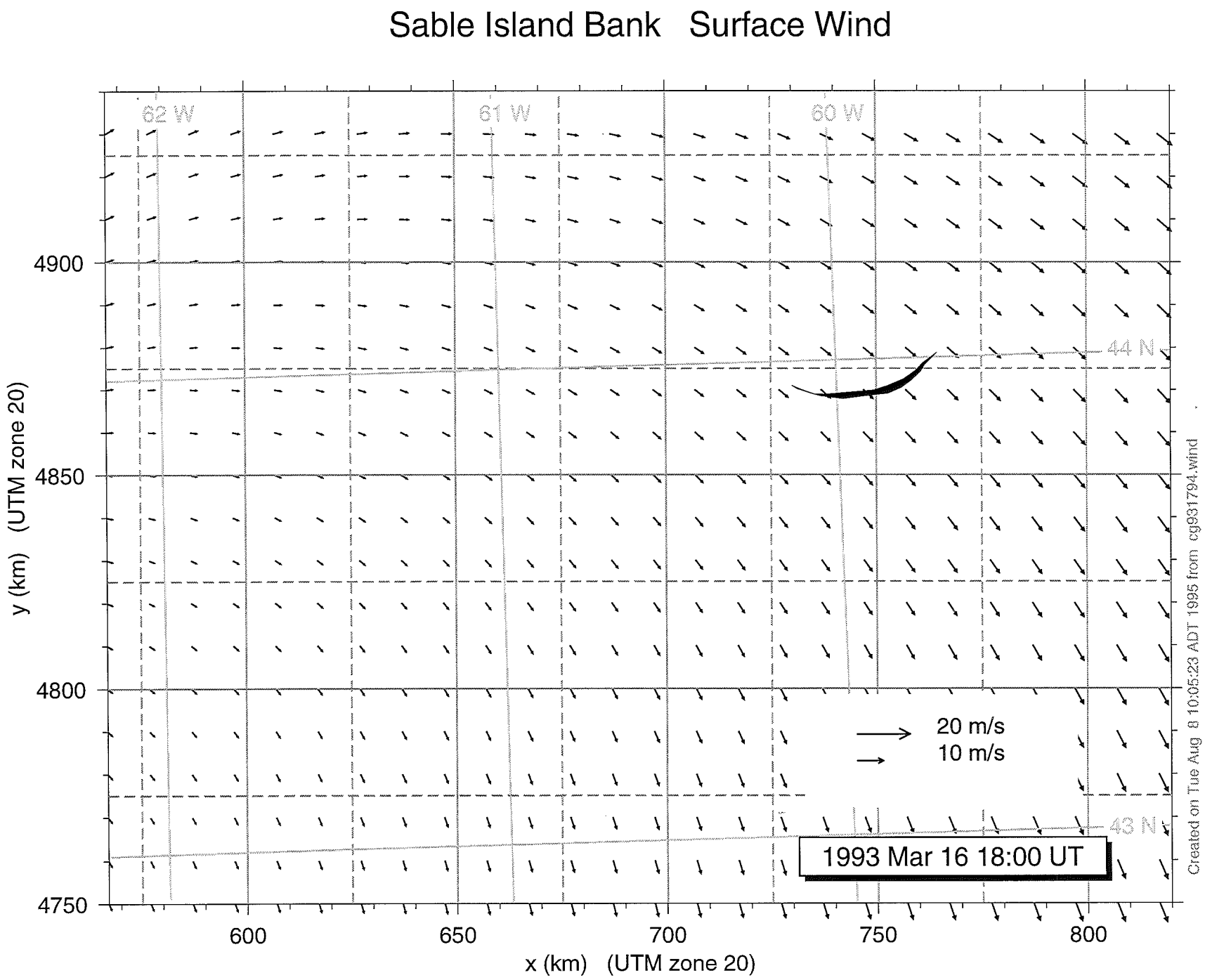

Sable Island Bank Surface Wind
62 W
61 W
60 W
44 N
43 N
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
20 m/s
10 m/s
1993 Mar 16 18:00 UT
Created on Tue Aug 8 10:05:23 ADT 1995 from cg931794.wind

Figure 5.11

Sable Island Bank Surface Wind

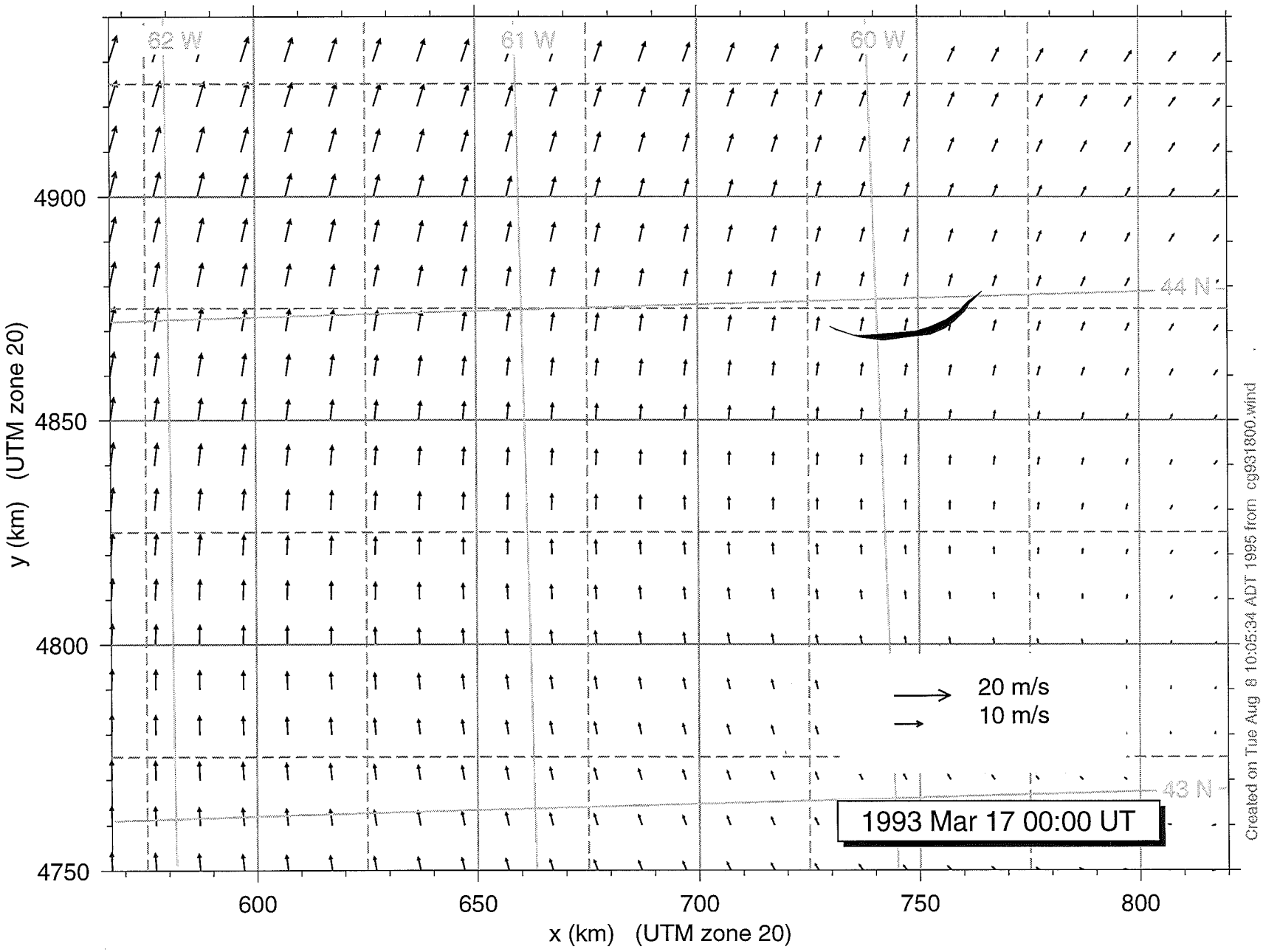


Figure 5.12

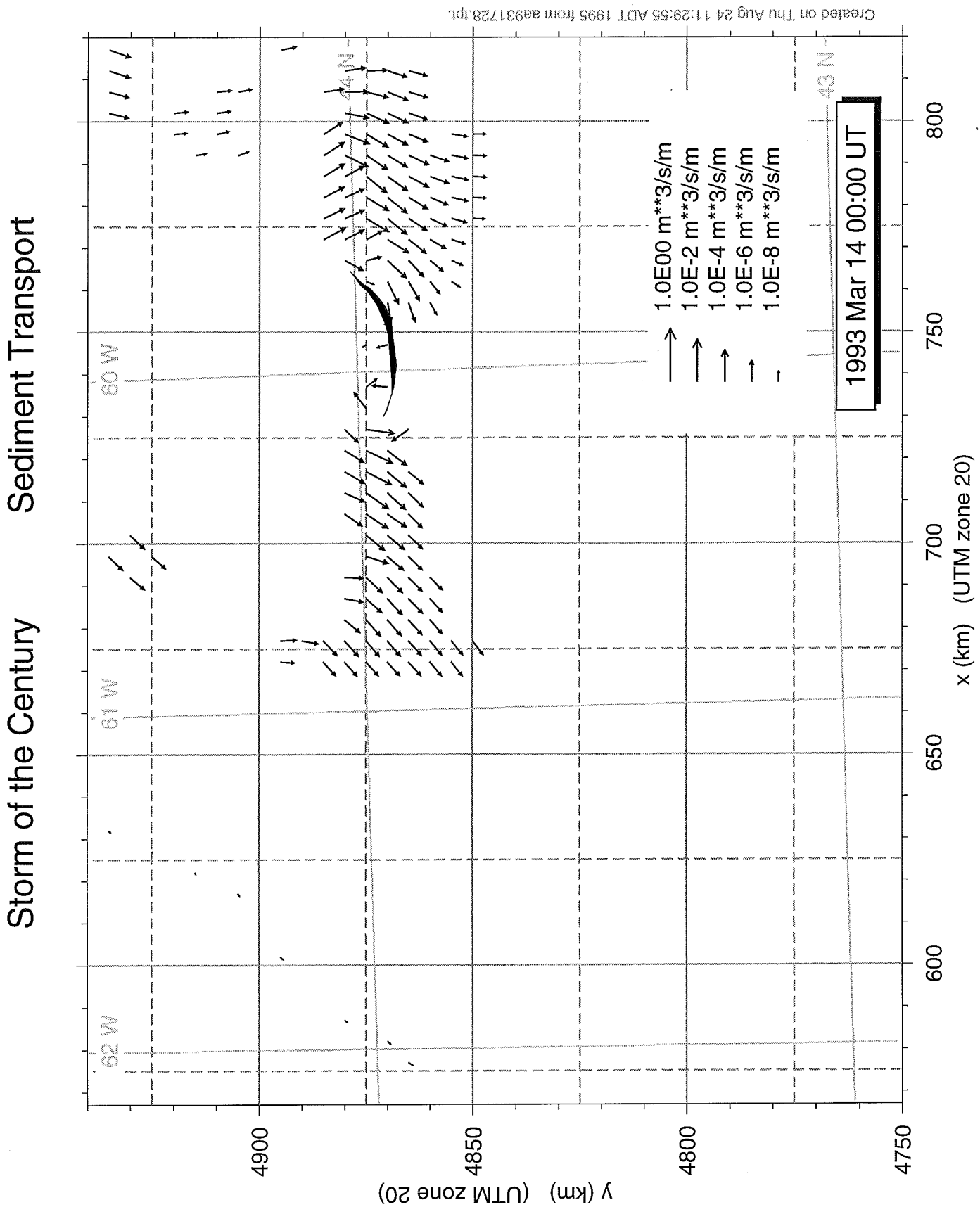

Storm of the Century
Sediment Transport
Created on Thu Aug 24 11:29:55 ADT 1995 from aa931728.tpt
1993 Mar 14 00:00 UT
1.0E00 m**3/s/m
1.0E-2 m**3/s/m
1.0E-4 m**3/s/m
1.0E-6 m**3/s/m
1.0E-8 m**3/s/m
62 W
61 W
60 W
44 N
43 N
600
650
700
750
800
x (km) (UTM zone 20)
4750
4800
4850
4900
y (km) (UTM zone 20)

Figure 6.1

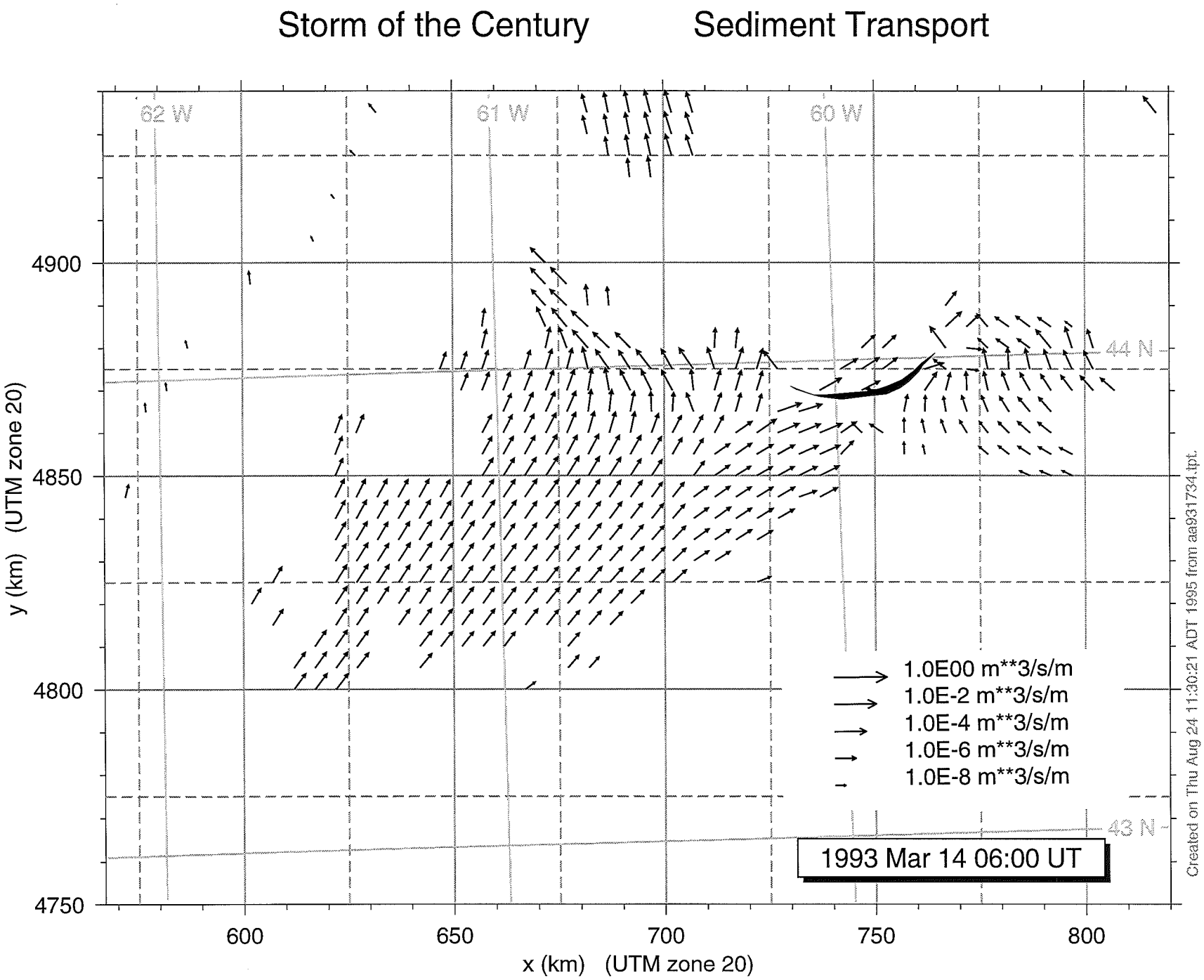

Storm of the Century
Sediment Transport
62 W
61 W
60 W
44 N
43 N
4900
4850
4800
4750
600
650
700
750
800
y (km) (UTM zone 20)
x (km) (UTM zone 20)
1.0E00 m**3/s/m
1.0E-2 m**3/s/m
1.0E-4 m**3/s/m
1.0E-6 m**3/s/m
1.0E-8 m**3/s/m
1993 Mar 14 06:00 UT
Created on Thu Aug 24 11:30:21 ADT 1995 from aa931734.tpt.

Figure 6.2

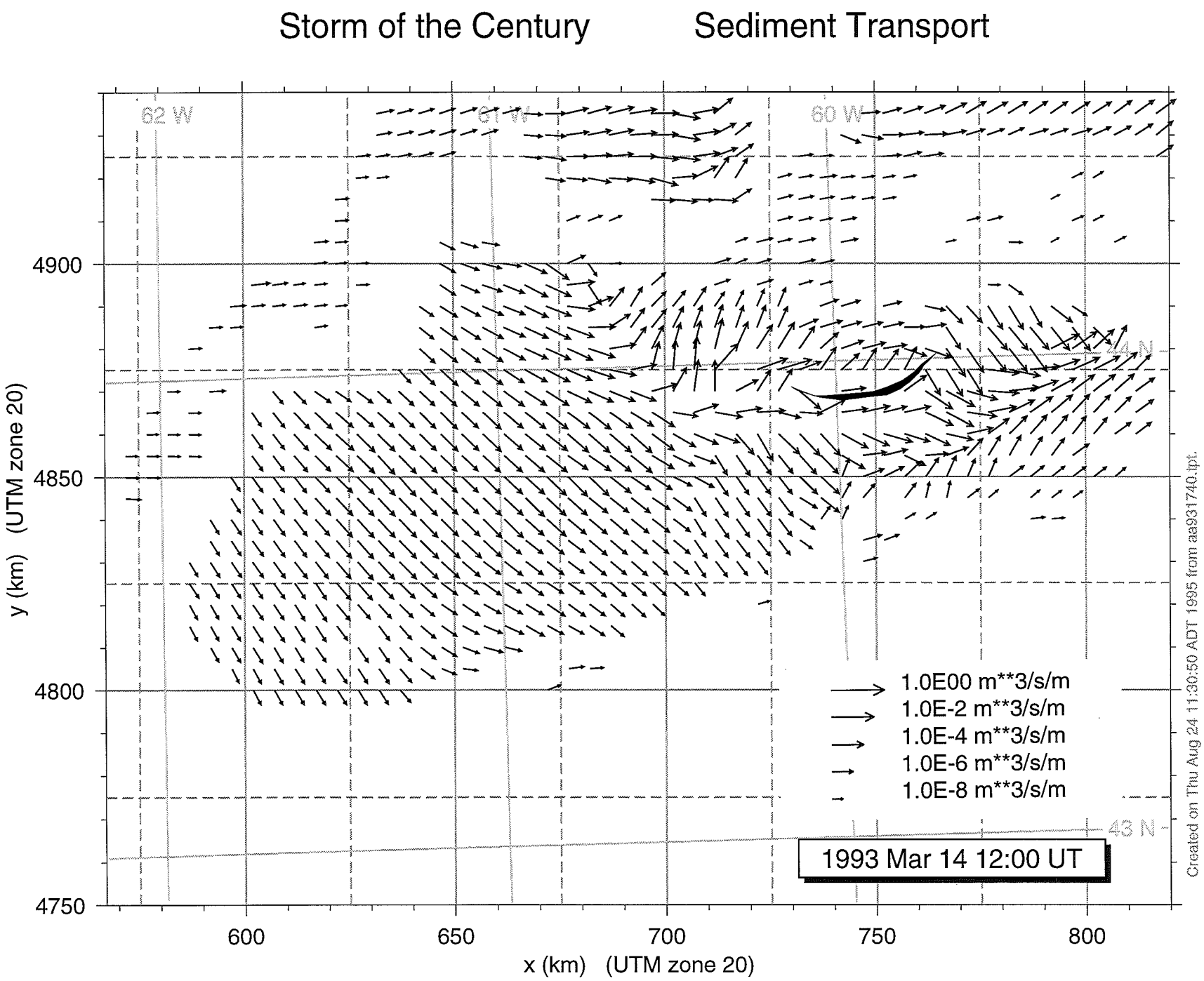

Storm of the Century
Sediment Transport
62 W
61 W
60 W
44 N
43 N
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
1.0E00 m**3/s/m
1.0E-2 m**3/s/m
1.0E-4 m**3/s/m
1.0E-6 m**3/s/m
1.0E-8 m**3/s/m
1993 Mar 14 12:00 UT
Created on Thu Aug 24 11:30:50 ADT 1995 from aa931740.lpt.

Figure 6.3

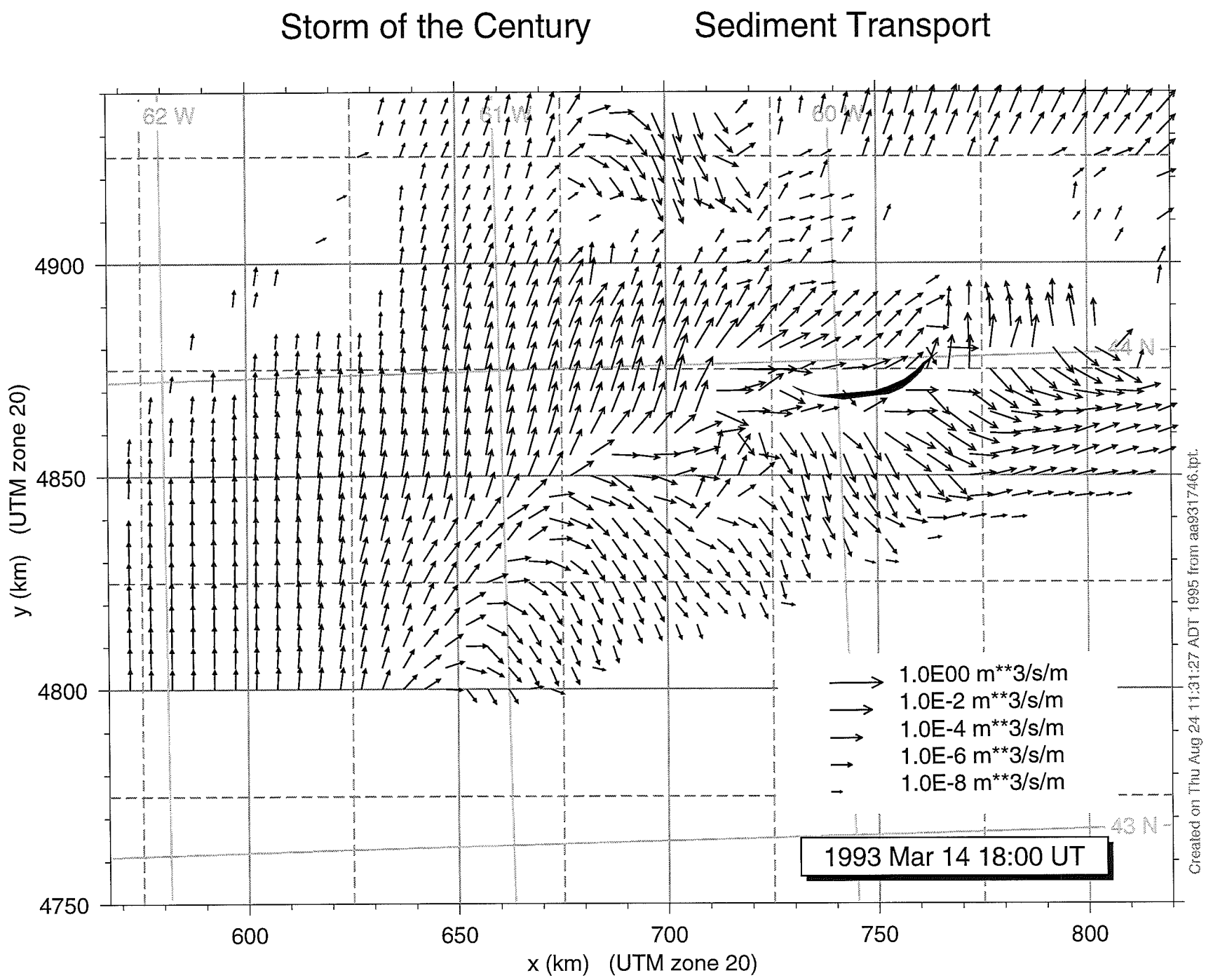

Storm of the Century
Sediment Transport
62 W
61 W
60 W
44 N
43 N
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
1.0E00 m**3/s/m
1.0E-2 m**3/s/m
1.0E-4 m**3/s/m
1.0E-6 m**3/s/m
1.0E-8 m**3/s/m
1993 Mar 14 18:00 UT
Created on Thu Aug 24 11:31:27 ADT 1995 from aa931746.tpt.

Figure 6.4

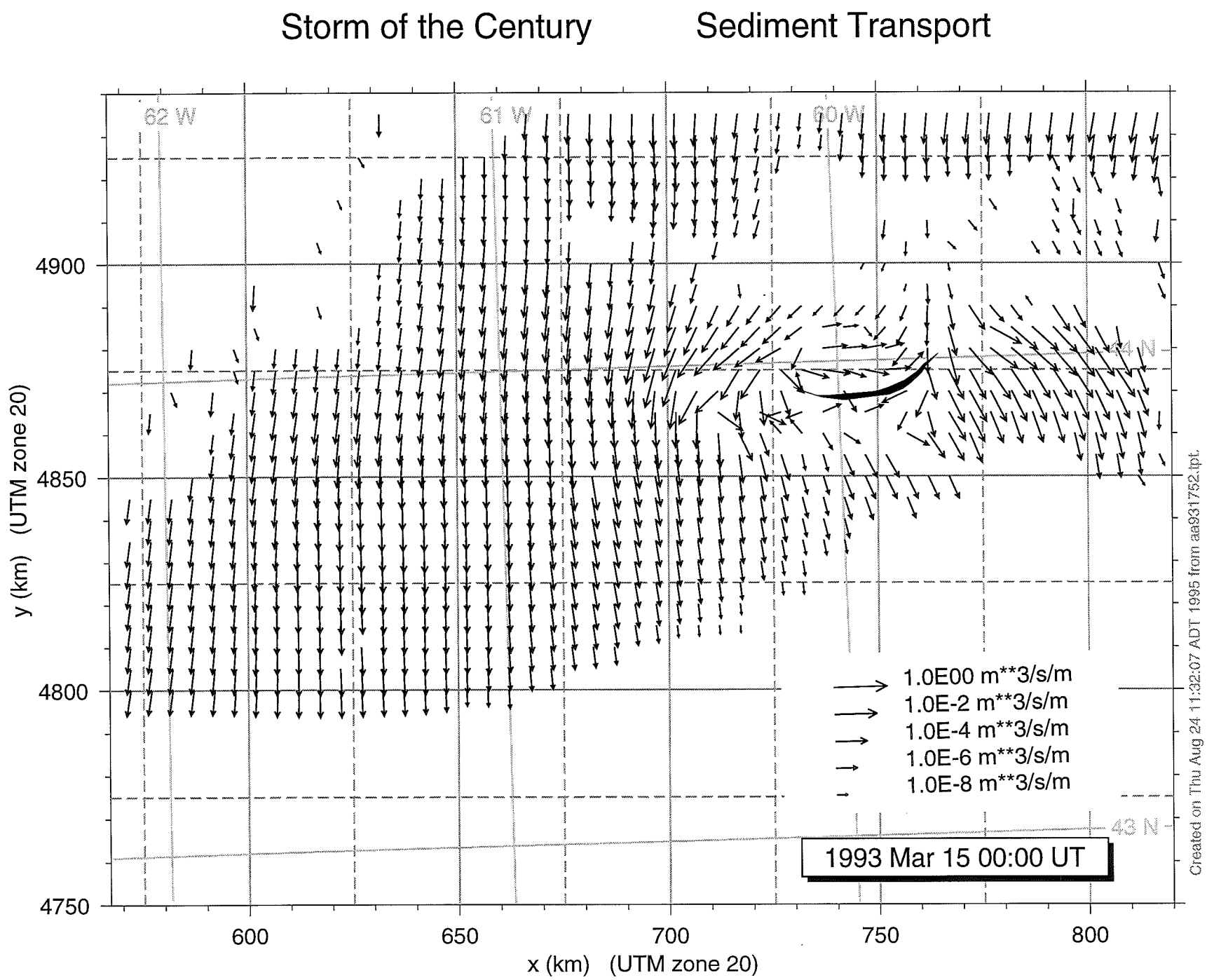

Storm of the Century
Sediment Transport
62 W
61 W
60 W
44 N
43 N
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
1.0E00 m**3/s/m
1.0E-2 m**3/s/m
1.0E-4 m**3/s/m
1.0E-6 m**3/s/m
1.0E-8 m**3/s/m
1993 Mar 15 00:00 UT
Created on Thu Aug 24 11:32:07 ADT 1995 from aa931752.lpt.

Figure 6.5

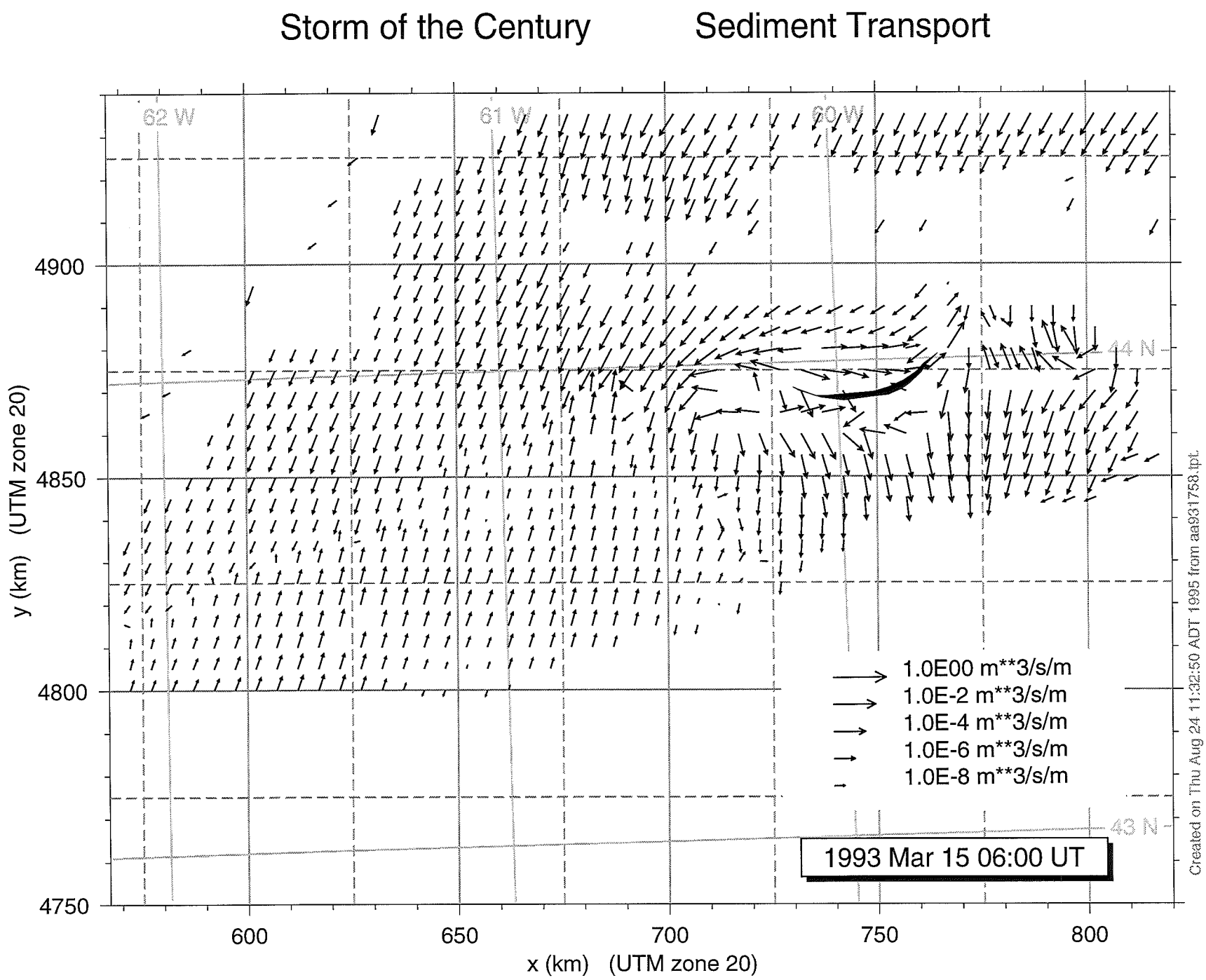

Storm of the Century
Sediment Transport
62 W
61 W
60 W
44 N
43 N
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
1.0E00 m**3/s/m
1.0E-2 m**3/s/m
1.0E-4 m**3/s/m
1.0E-6 m**3/s/m
1.0E-8 m**3/s/m
1993 Mar 15 06:00 UT
Created on Thu Aug 24 11:32:50 ADT 1995 from aa931758.tpt.

Figure 6.6

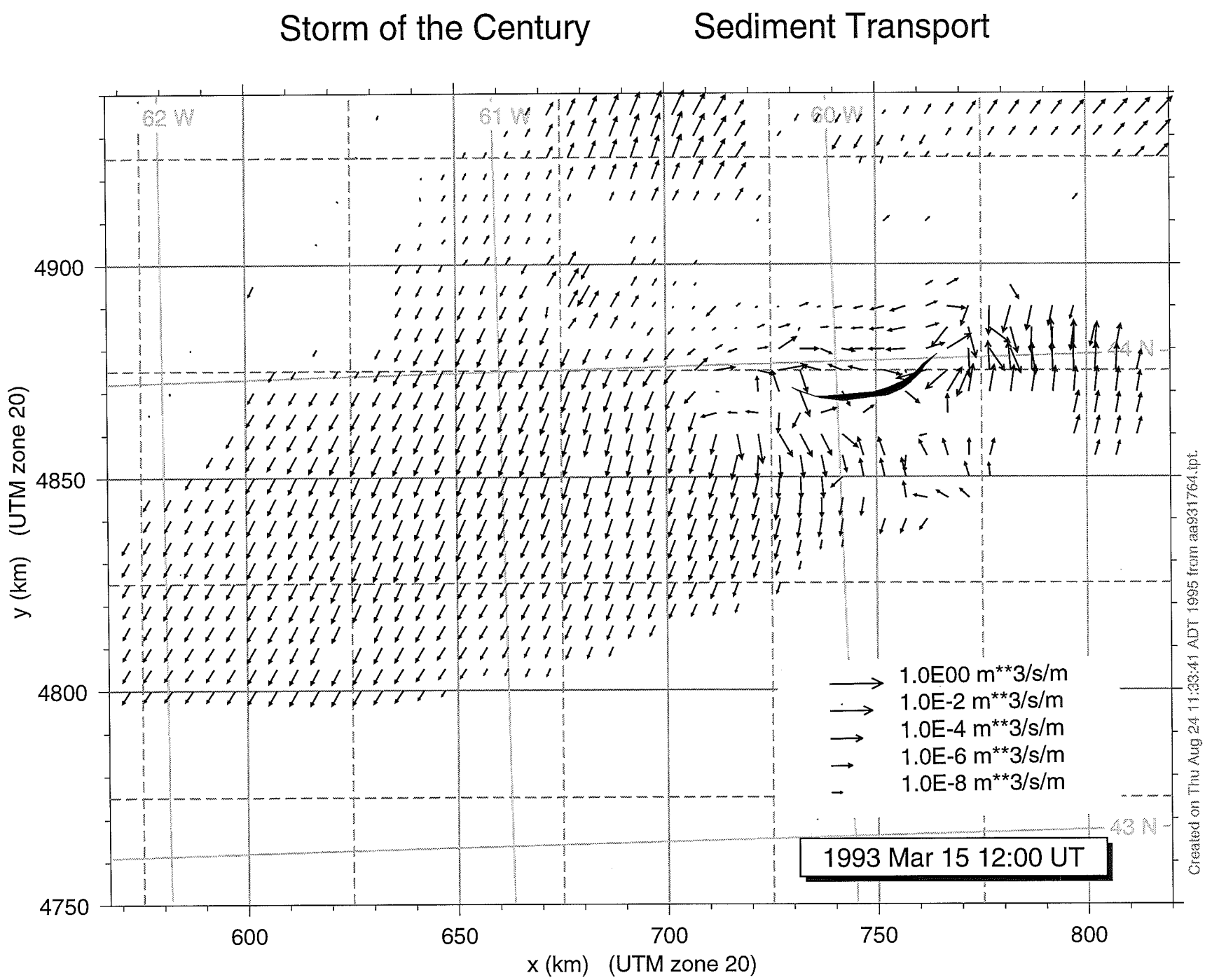

Storm of the Century
Sediment Transport
62 W
61 W
60 W
44 N
43 N
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
1.0E00 m**3/s/m
1.0E-2 m**3/s/m
1.0E-4 m**3/s/m
1.0E-6 m**3/s/m
1.0E-8 m**3/s/m
1993 Mar 15 12:00 UT
Created on Thu Aug 24 11:33:41 ADT 1995 from aa931764.ipt.

Figure 6.7

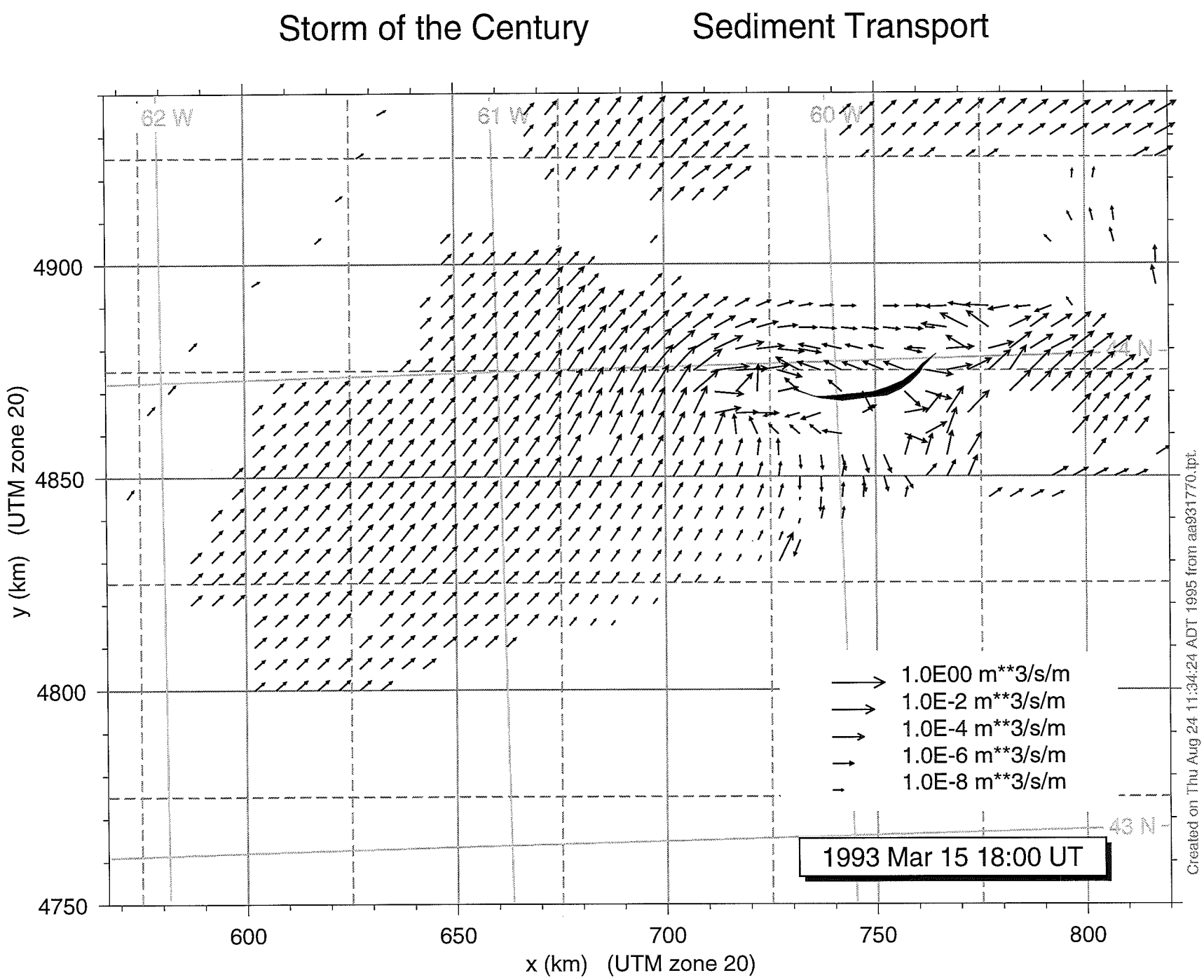

Storm of the Century
Sediment Transport
62 W
61 W
60 W
44 N
43 N
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
1.0E00 m**3/s/m
1.0E-2 m**3/s/m
1.0E-4 m**3/s/m
1.0E-6 m**3/s/m
1.0E-8 m**3/s/m
1993 Mar 15 18:00 UT
Created on Thu Aug 24 11:34:24 ADT 1995 from aa931770.tpt

Figure 6.8

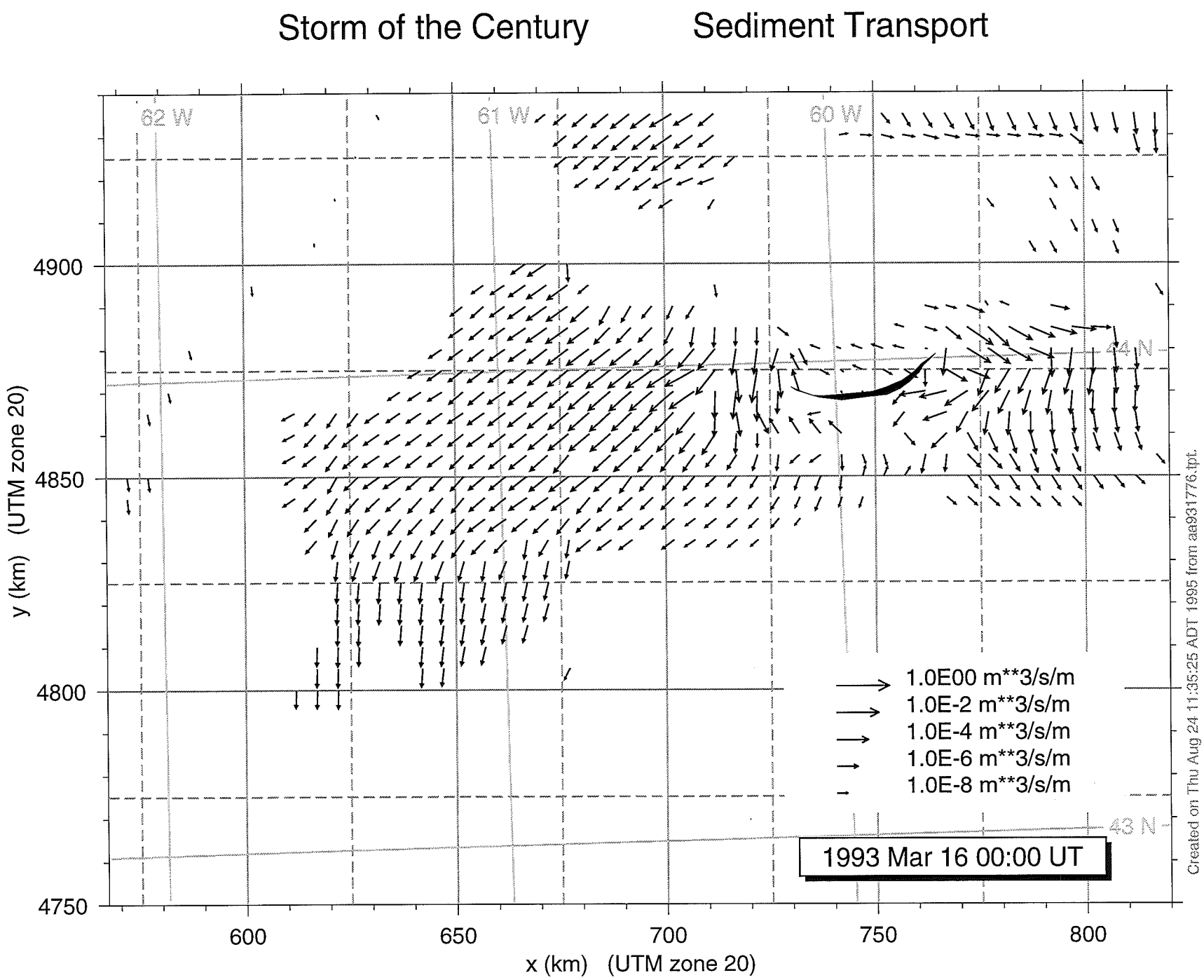

Storm of the Century
Sediment Transport
62 W
61 W
60 W
44 N
43 N
4900
4850
4800
4750
600
650
700
750
800
y (km) (UTM zone 20)
x (km) (UTM zone 20)
1.0E00 m**3/s/m
1.0E-2 m**3/s/m
1.0E-4 m**3/s/m
1.0E-6 m**3/s/m
1.0E-8 m**3/s/m
1993 Mar 16 00:00 UT
Created on Thu Aug 24 11:35:25 ADT 1995 from aa931776.tpt.

Figure 6.9

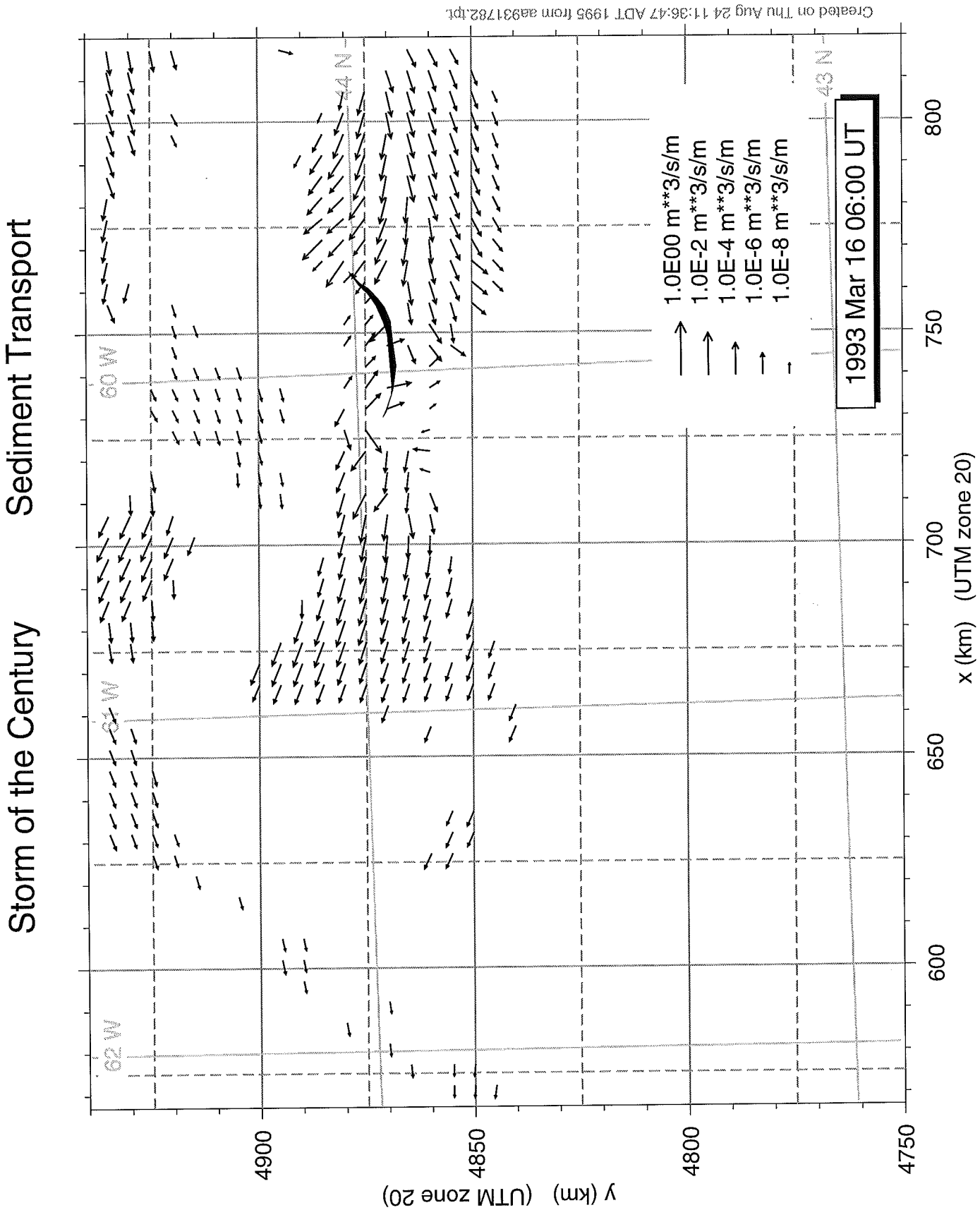

Storm of the Century
Sediment Transport
Created on Thu Aug 24 11:36:47 ADT 1995 from aa9317821.tpt
1993 Mar 16 06:00 UT
1.0E00 m**3/s/m
1.0E-2 m**3/s/m
1.0E-4 m**3/s/m
1.0E-6 m**3/s/m
1.0E-8 m**3/s/m
62 W
64 W
60 W
44 N
43 N
600
650
700
750
800
x (km) (UTM zone 20)
4900
4850
4800
4750
y (km) (UTM zone 20)

Figure 6.10

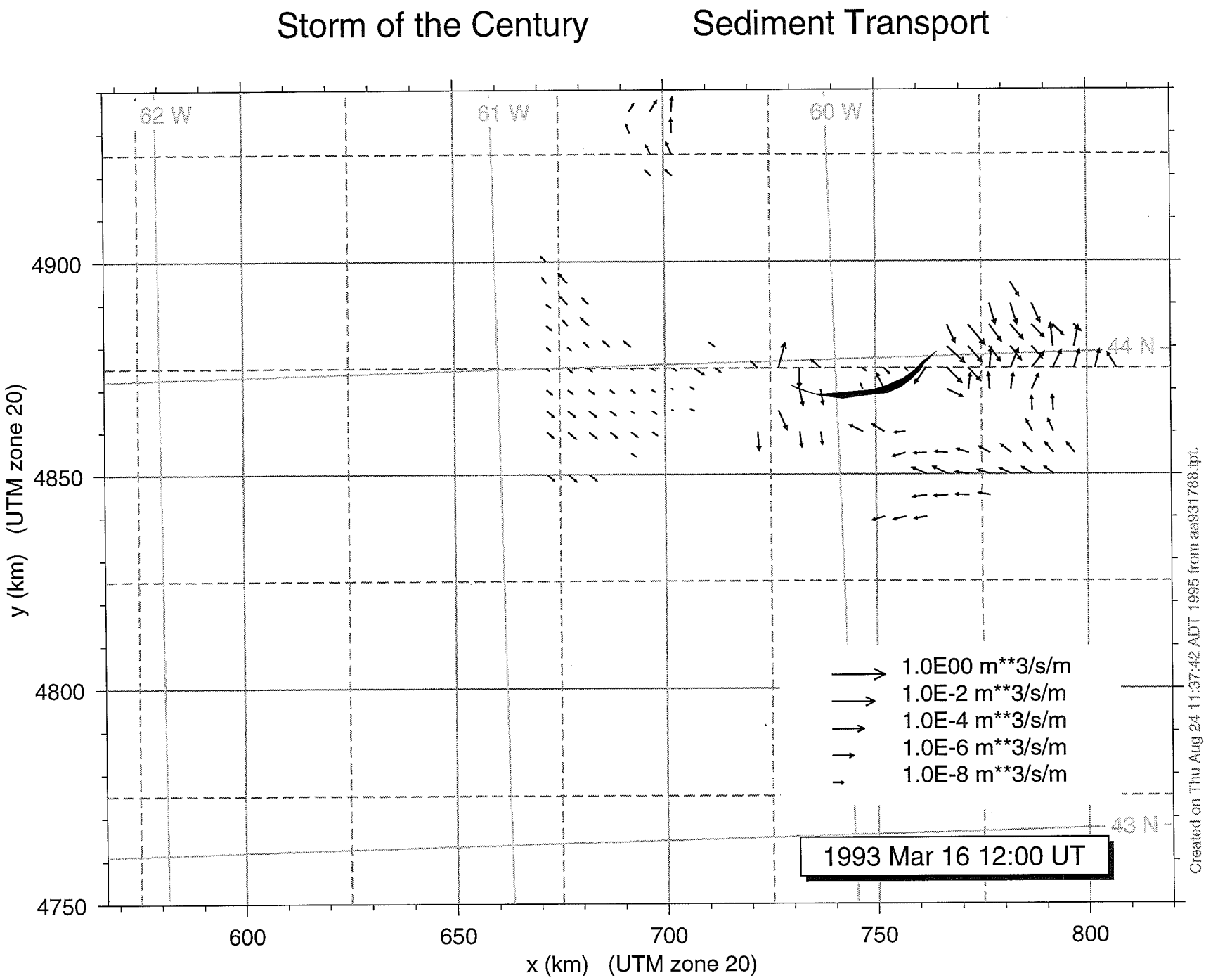

Storm of the Century
Sediment Transport
62 W
61 W
60 W
44 N
43 N
4900
4850
4800
4750
600
650
700
750
800
y (km) (UTM zone 20)
x (km) (UTM zone 20)
1.0E00 m**3/s/m
1.0E-2 m**3/s/m
1.0E-4 m**3/s/m
1.0E-6 m**3/s/m
1.0E-8 m**3/s/m
1993 Mar 16 12:00 UT
Created on Thu Aug 24 11:37:42 ADT 1995 from aa931788.lpt.

Figure 6.11

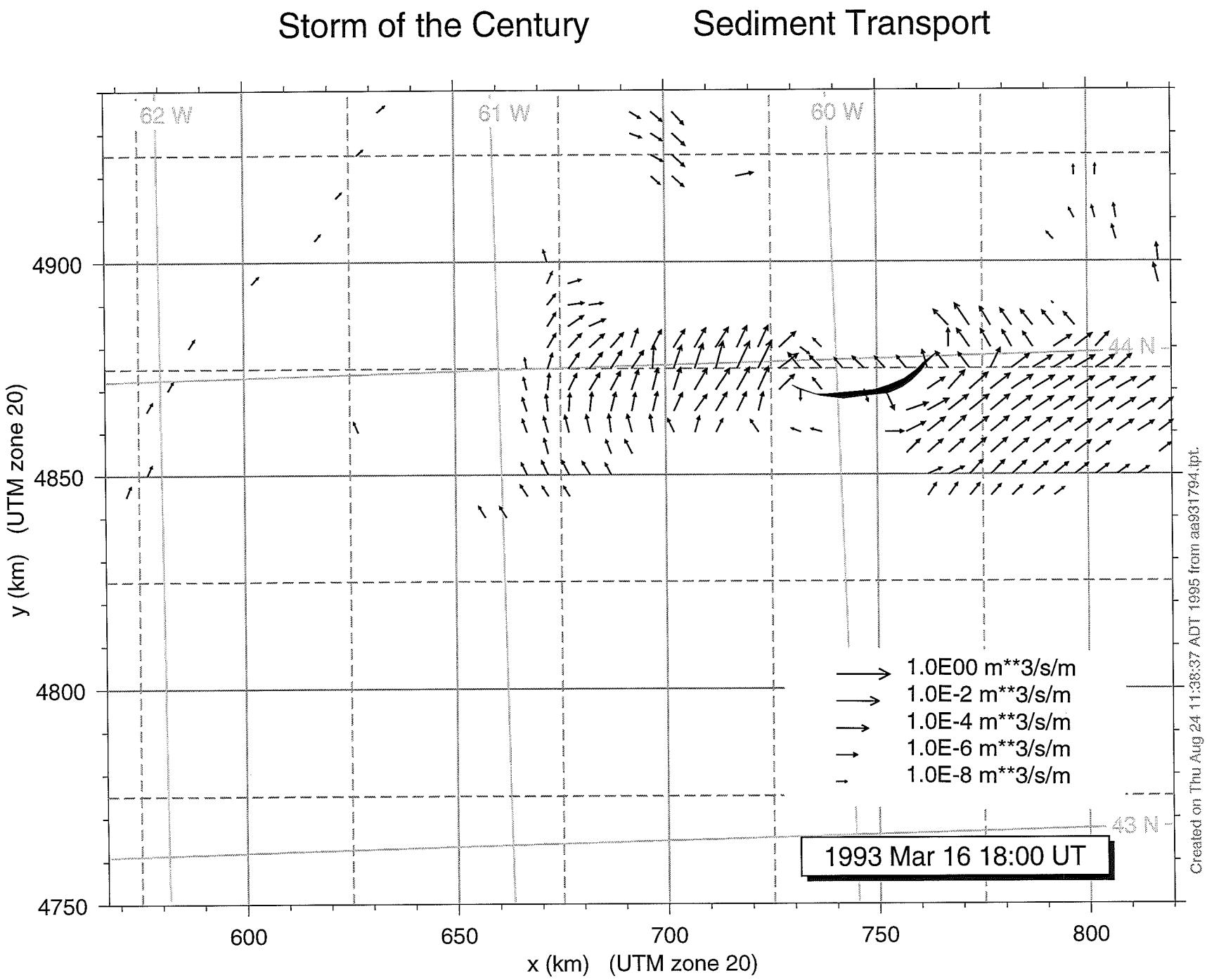

Storm of the Century
Sediment Transport
62 W
61 W
60 W
44 N
43 N
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
1.0E00 m**3/s/m
1.0E-2 m**3/s/m
1.0E-4 m**3/s/m
1.0E-6 m**3/s/m
1.0E-8 m**3/s/m
1993 Mar 16 18:00 UT
Created on Thu Aug 24 11:38:37 ADT 1995 from aa931794.tpt.

Figure 6.12

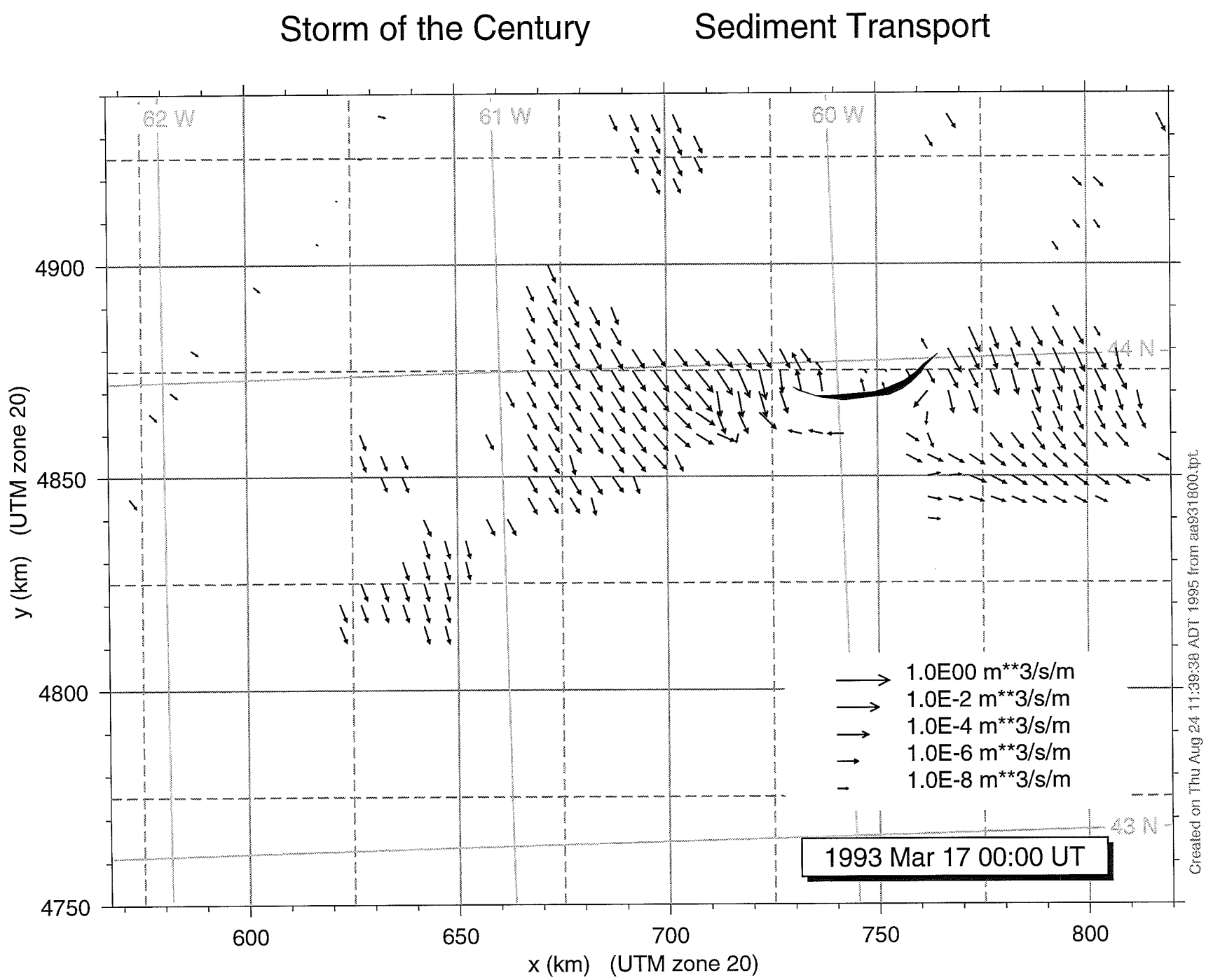

Storm of the Century
Sediment Transport
62 W
61 W
60 W
44 N
43 N
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
1.0E00 m**3/s/m
1.0E-2 m**3/s/m
1.0E-4 m**3/s/m
1.0E-6 m**3/s/m
1.0E-8 m**3/s/m
1993 Mar 17 00:00 UT
Created on Thu Aug 24 11:39:38 ADT 1995 from aa931800.tpt.

Figure 6.13

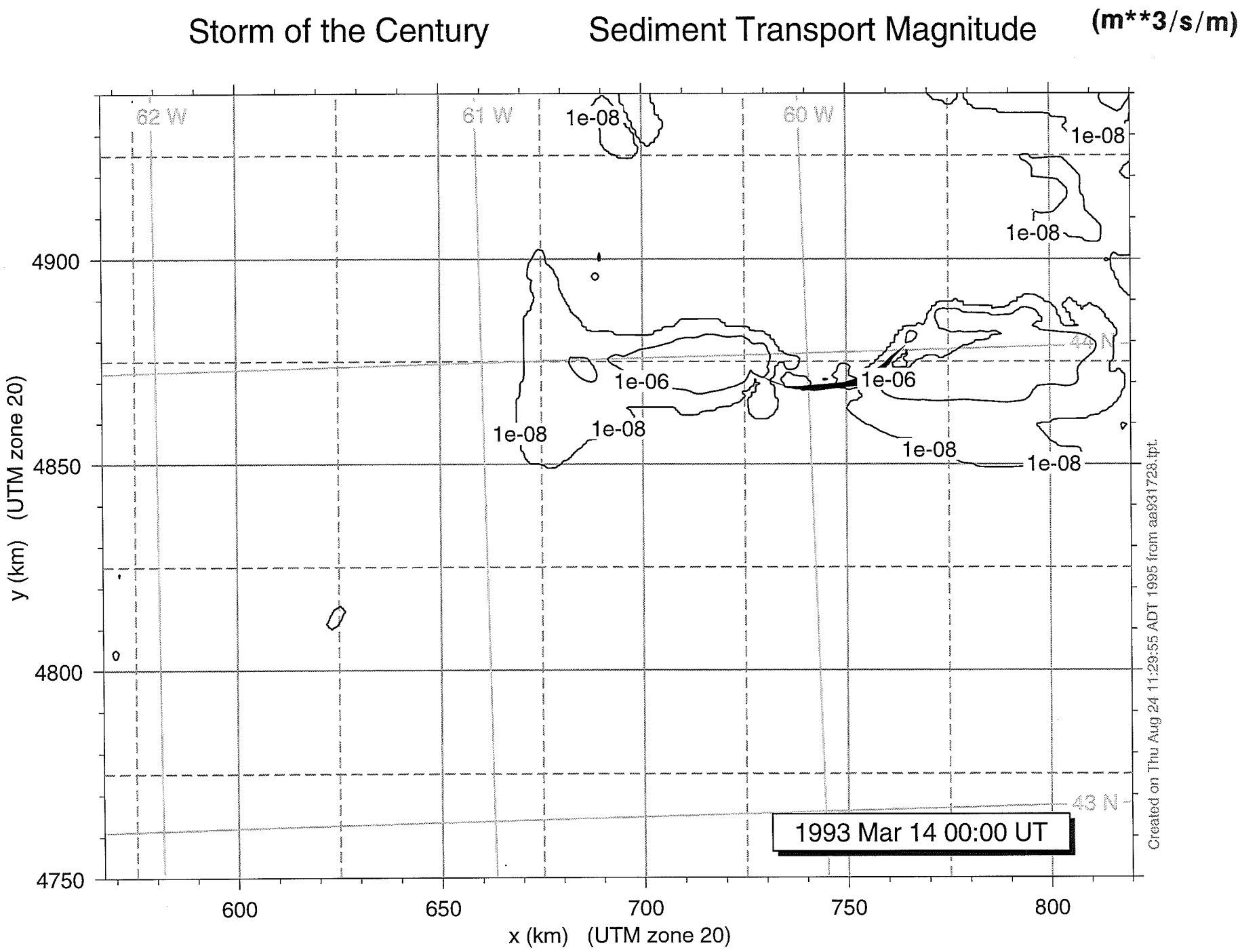

Storm of the Century
Sediment Transport Magnitude
(m**3/s/m)
62 W
61 W
60 W
44 N
43 N
1e-08
1e-06
1993 Mar 14 00:00 UT
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
Created on Thu Aug 24 11:29:55 ADT 1995 from aa931728.tpt.

Figure 7.1

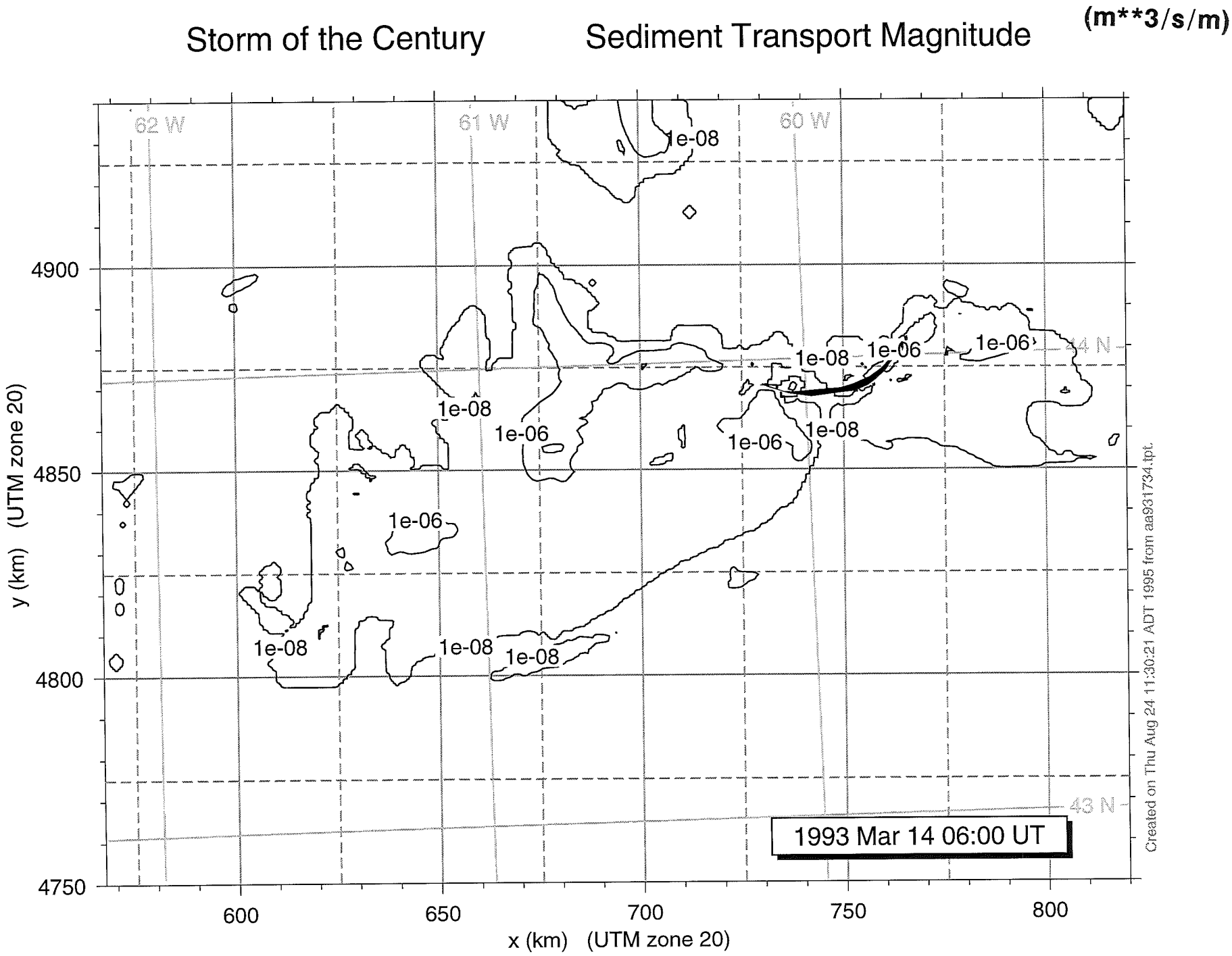

Storm of the Century
Sediment Transport Magnitude
(m**3/s/m)
62 W
61 W
60 W
44 N
43 N
1e-08
1e-06
1e-08
1e-06
1e-06
1e-08
1e-08
1e-06
1e-06
1e-08
1e-08
1e-08
1993 Mar 14 06:00 UT
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
Created on Thu Aug 24 11:30:21 ADT 1995 from aa931734.tpt.

Figure 7.2

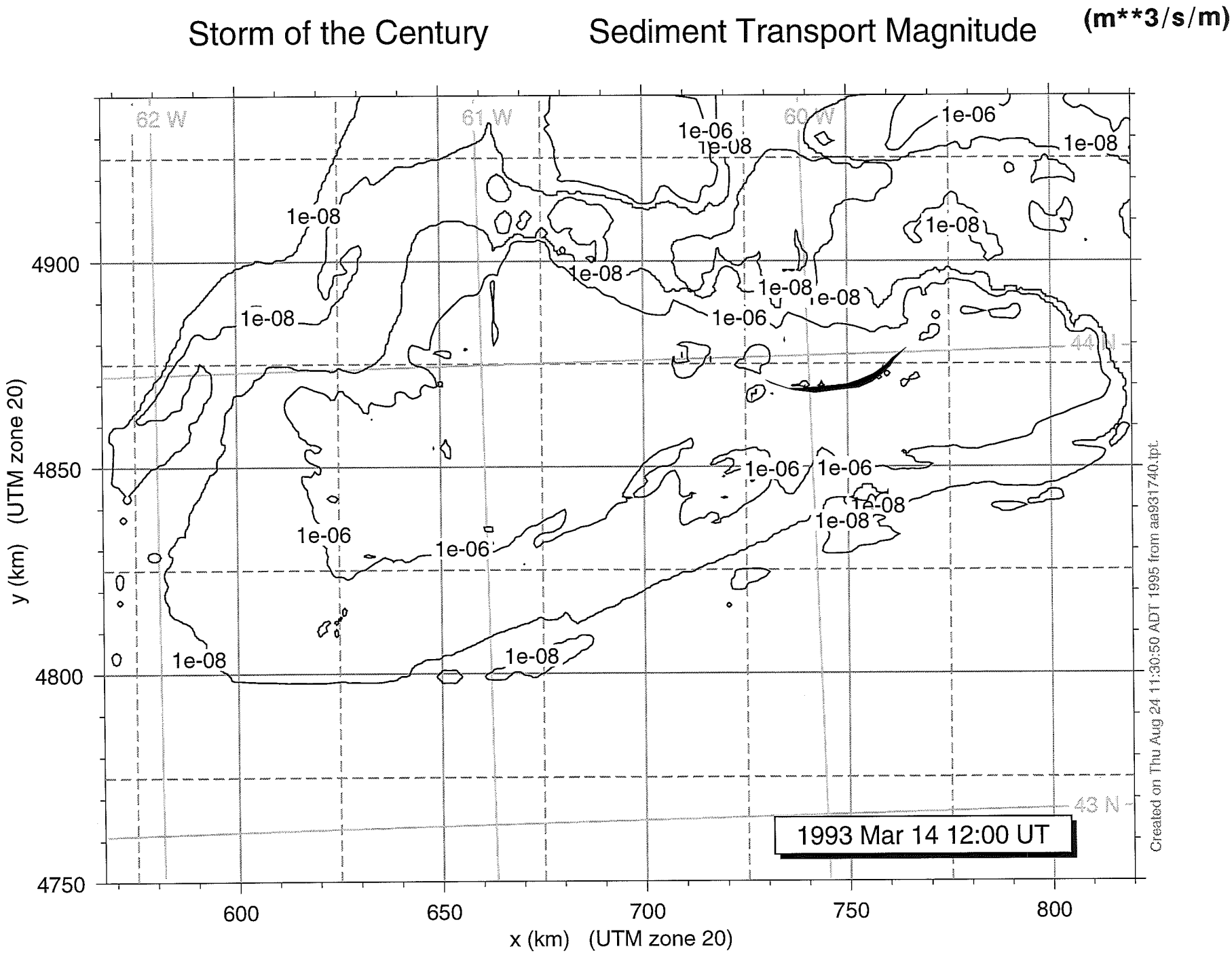

Storm of the Century
Sediment Transport Magnitude
(m**3/s/m)
62 W
61 W
60 W
44 N
43 N
1e-06
1e-08
1993 Mar 14 12:00 UT
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
Created on Thu Aug 24 11:30:50 ADT 1995 from aa931740.tpt

Figure 7.3

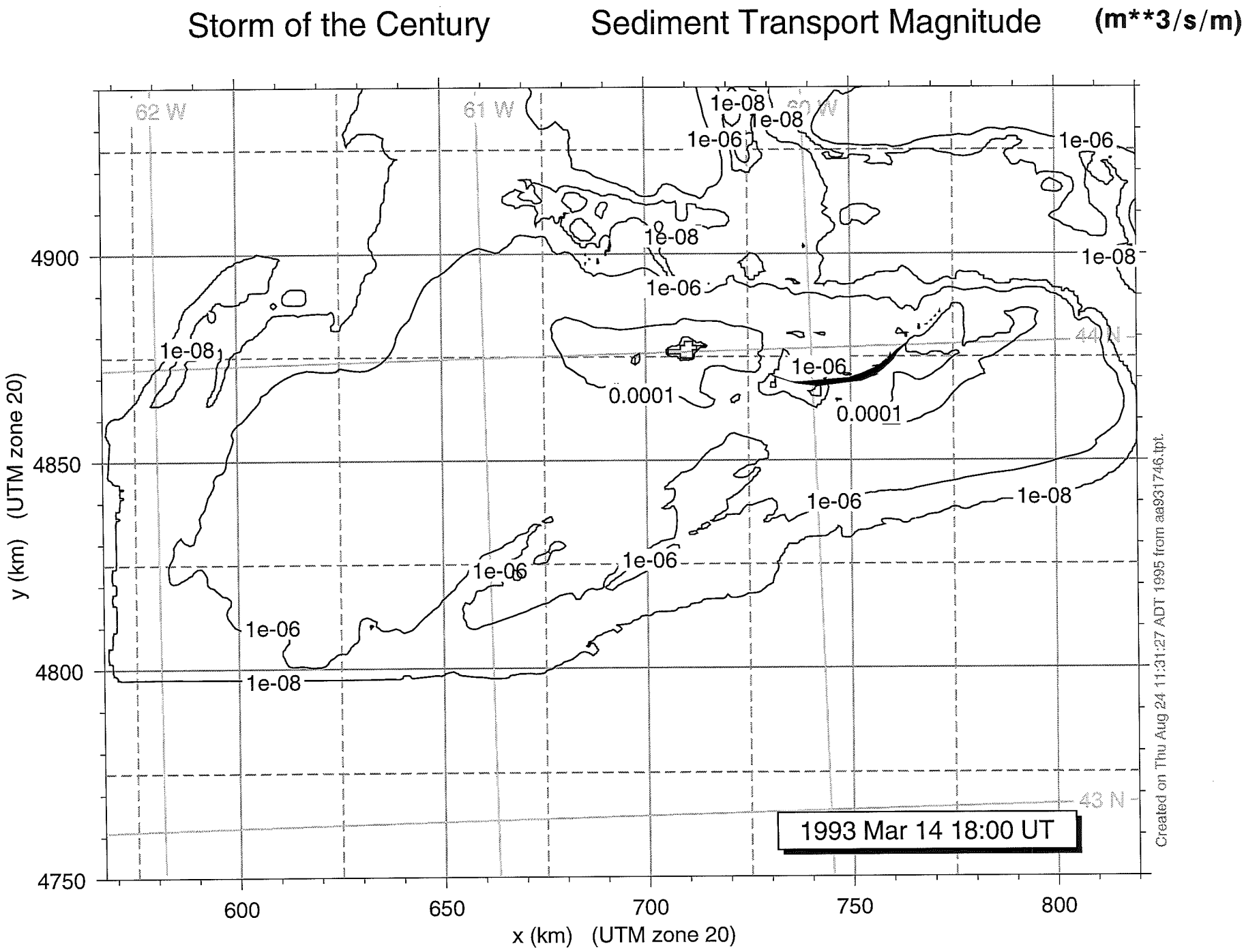

Storm of the Century
Sediment Transport Magnitude
(m**3/s/m)
62 W
61 W
60 W
44 N
43 N
1e-08
1e-06
0.0001
1993 Mar 14 18:00 UT
4900
4850
4800
4750
600
650
700
750
800
y (km) (UTM zone 20)
x (km) (UTM zone 20)
Created on Thu Aug 24 11:31:27 ADT 1995 from aa931746.tpt.

Figure 7.4

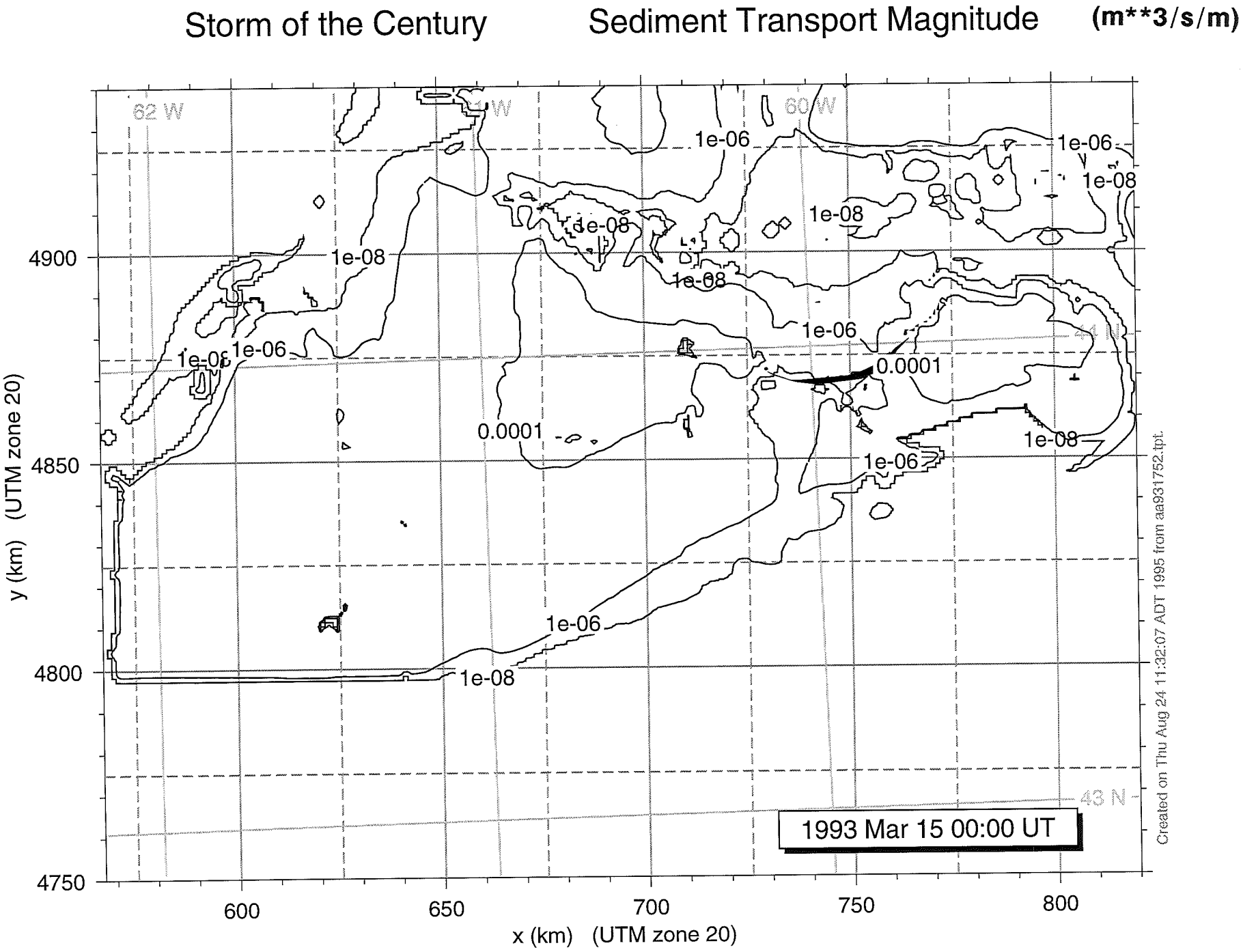

Storm of the Century
Sediment Transport Magnitude
(m**3/s/m)
62 W
60 W
44 N
43 N
1e-06
1e-08
0.0001
4900
4850
4800
4750
600
650
700
750
800
y (km) (UTM zone 20)
x (km) (UTM zone 20)
1993 Mar 15 00:00 UT
Created on Thu Aug 24 11:32:07 ADT 1995 from aa931752.tpl.

Figure 7.5

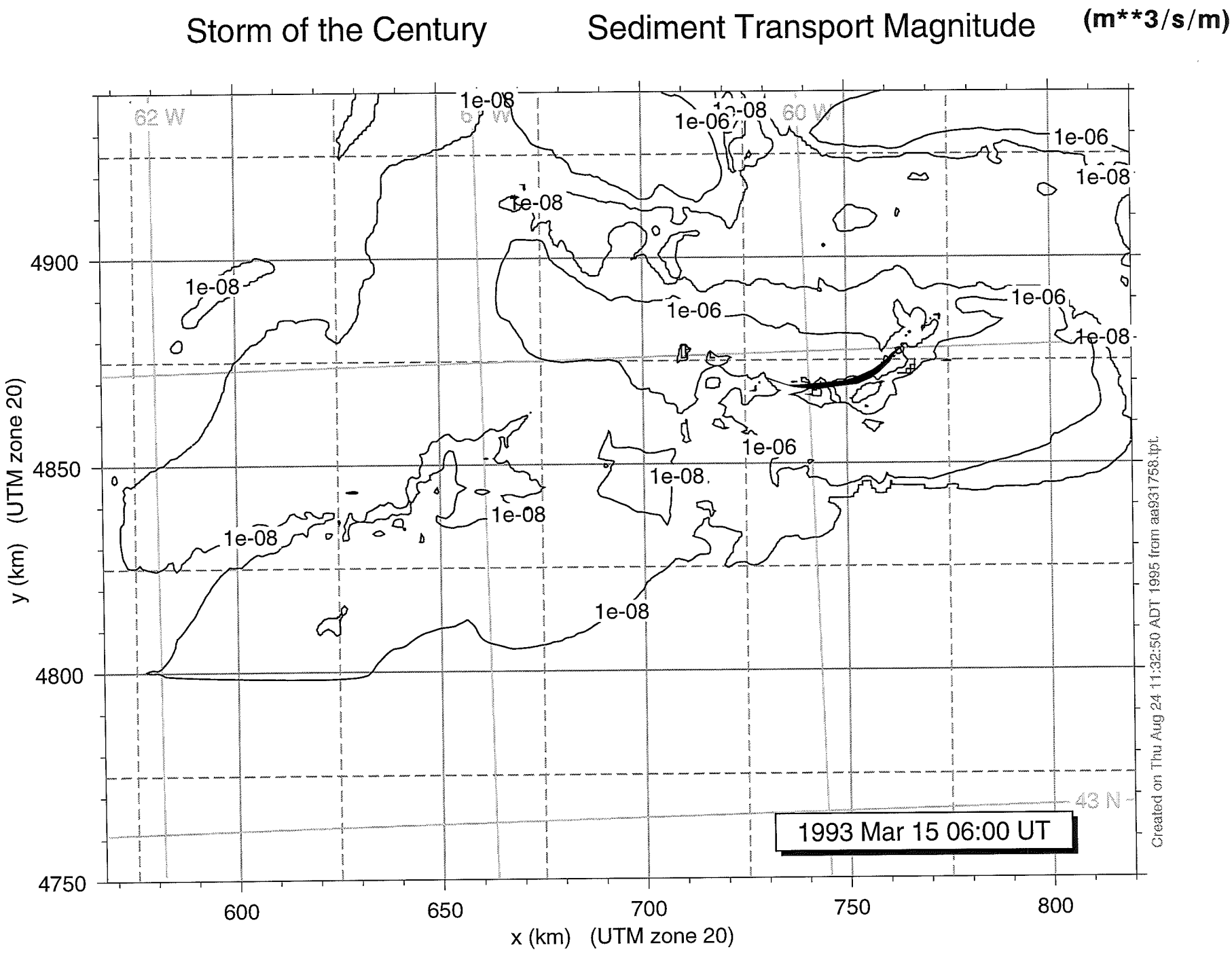

Storm of the Century
Sediment Transport Magnitude
(m**3/s/m)
62 W
60 W
43 N
1e-08
1e-06
1993 Mar 15 06:00 UT
4900
4850
4800
4750
600
650
700
750
800
y (km) (UTM zone 20)
x (km) (UTM zone 20)
Created on Thu Aug 24 11:32:50 ADT 1995 from aa931758.tpt.

Figure 7.6

Storm of the Century Sediment Transport Magnitude (m**3/s/m)

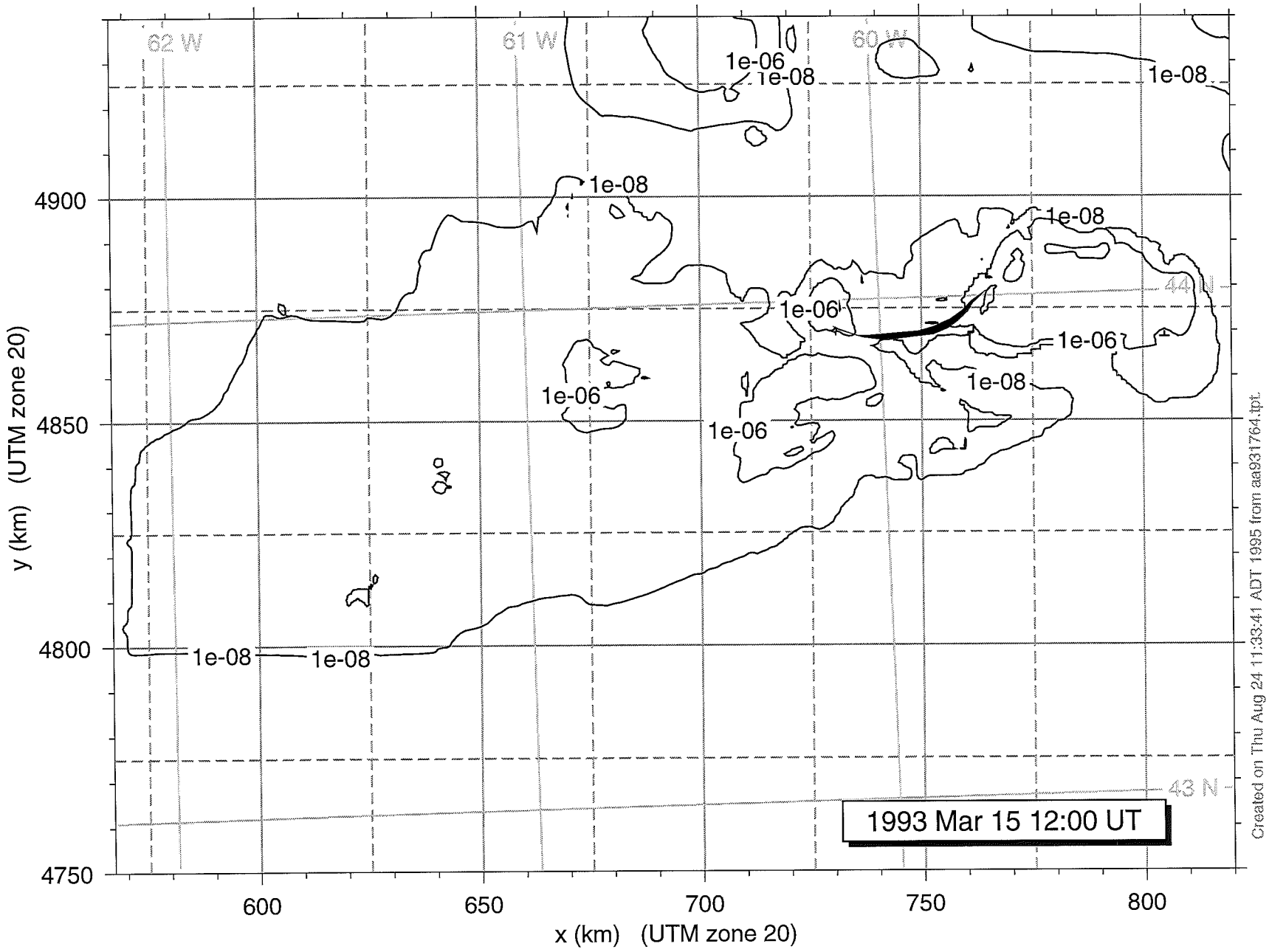


Figure 7.7

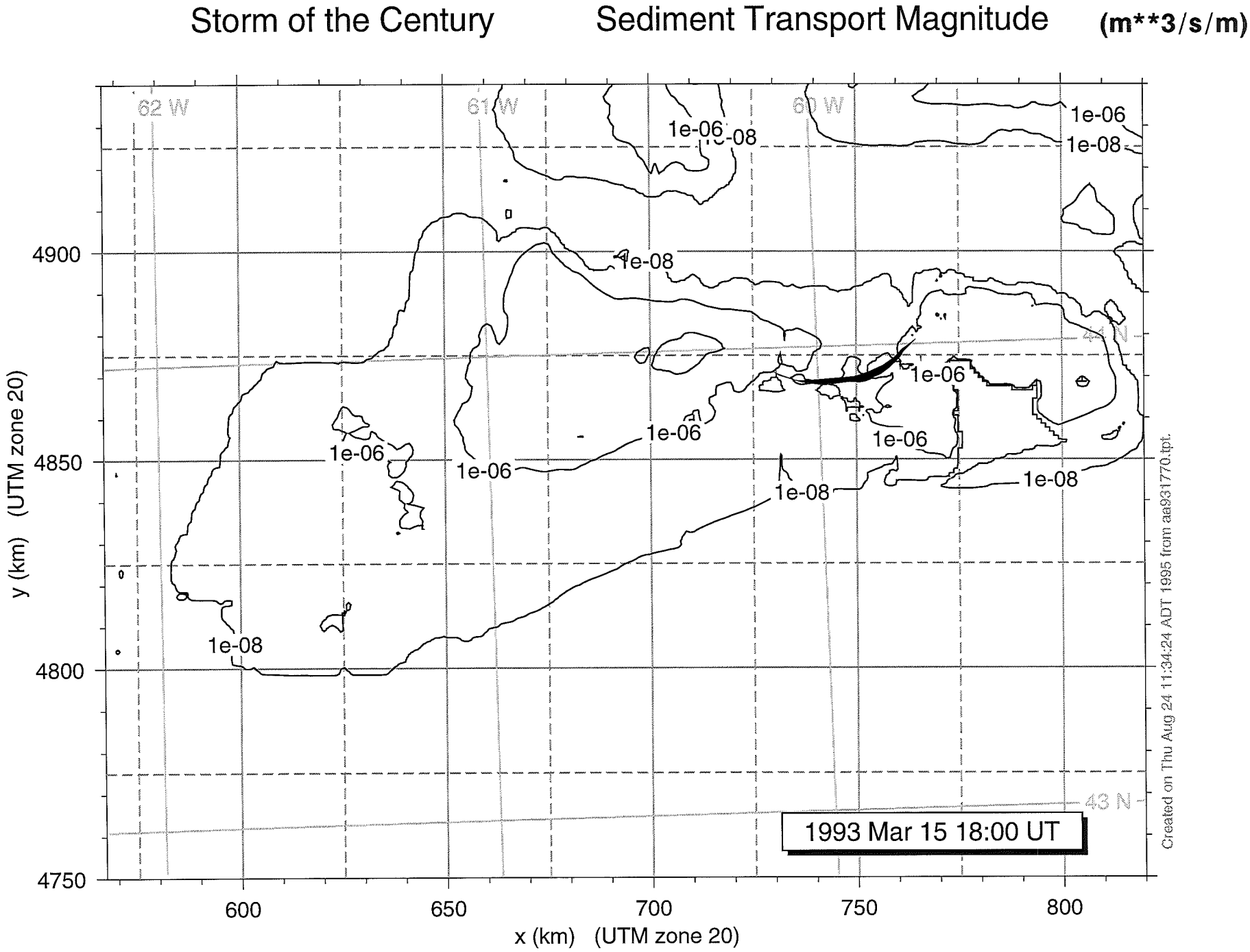

Storm of the Century
Sediment Transport Magnitude
(m**3/s/m)
62 W
61 W
60 W
44 N
43 N
1e-06
1e-08
1993 Mar 15 18:00 UT
4900
4850
4800
4750
600
650
700
750
800
y (km) (UTM zone 20)
x (km) (UTM zone 20)
Created on Thu Aug 24 11:34:24 ADT 1995 from aa931770.tpt.

Figure 7.8

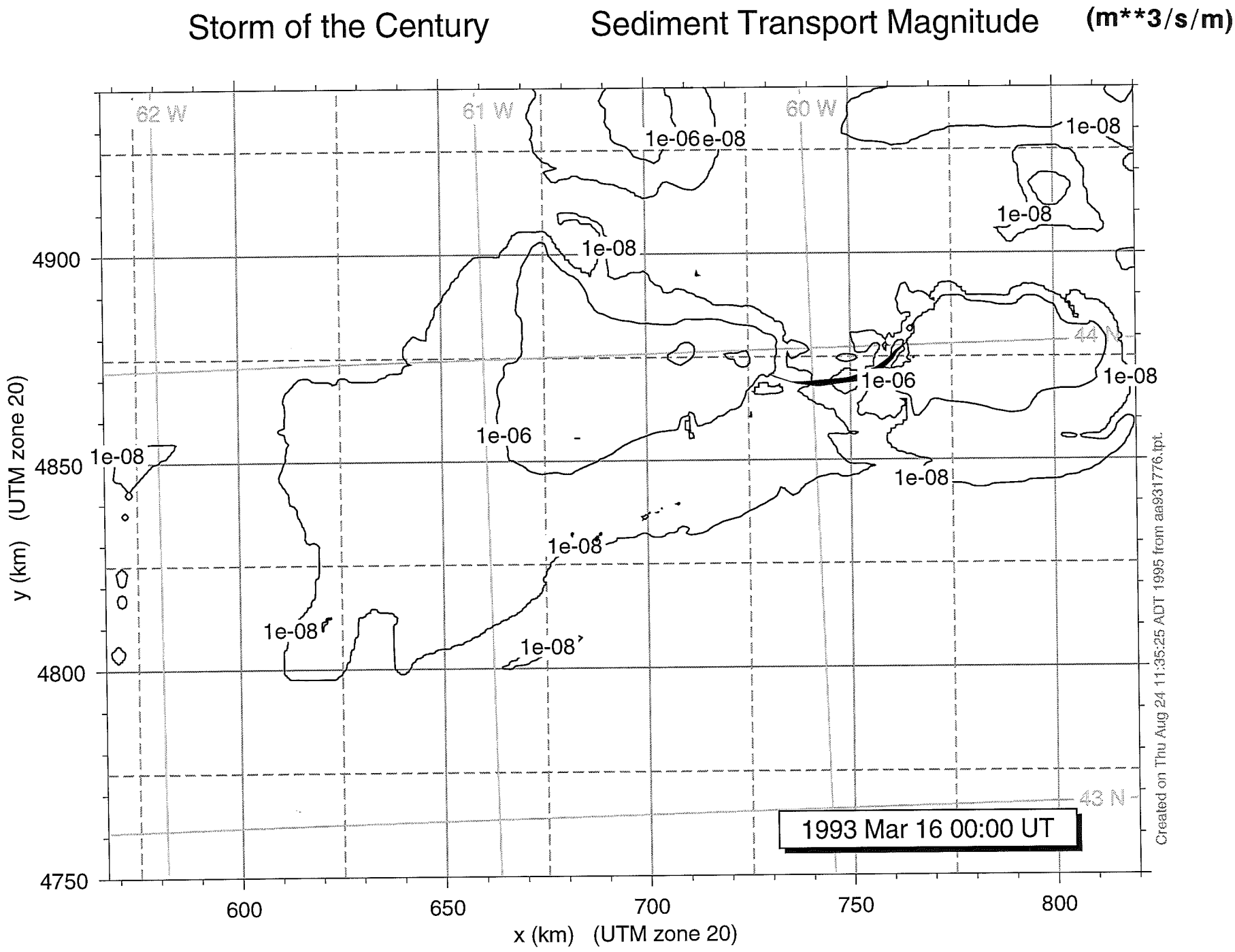

Storm of the Century
Sediment Transport Magnitude
(m**3/s/m)
62 W
61 W
60 W
44 N
43 N
1e-06e-08
1e-08
1e-06
1993 Mar 16 00:00 UT
4900
4850
4800
4750
600
650
700
750
800
y (km) (UTM zone 20)
x (km) (UTM zone 20)
Created on Thu Aug 24 11:35:25 ADT 1995 from aa931776.tpt.

Figure 7.9

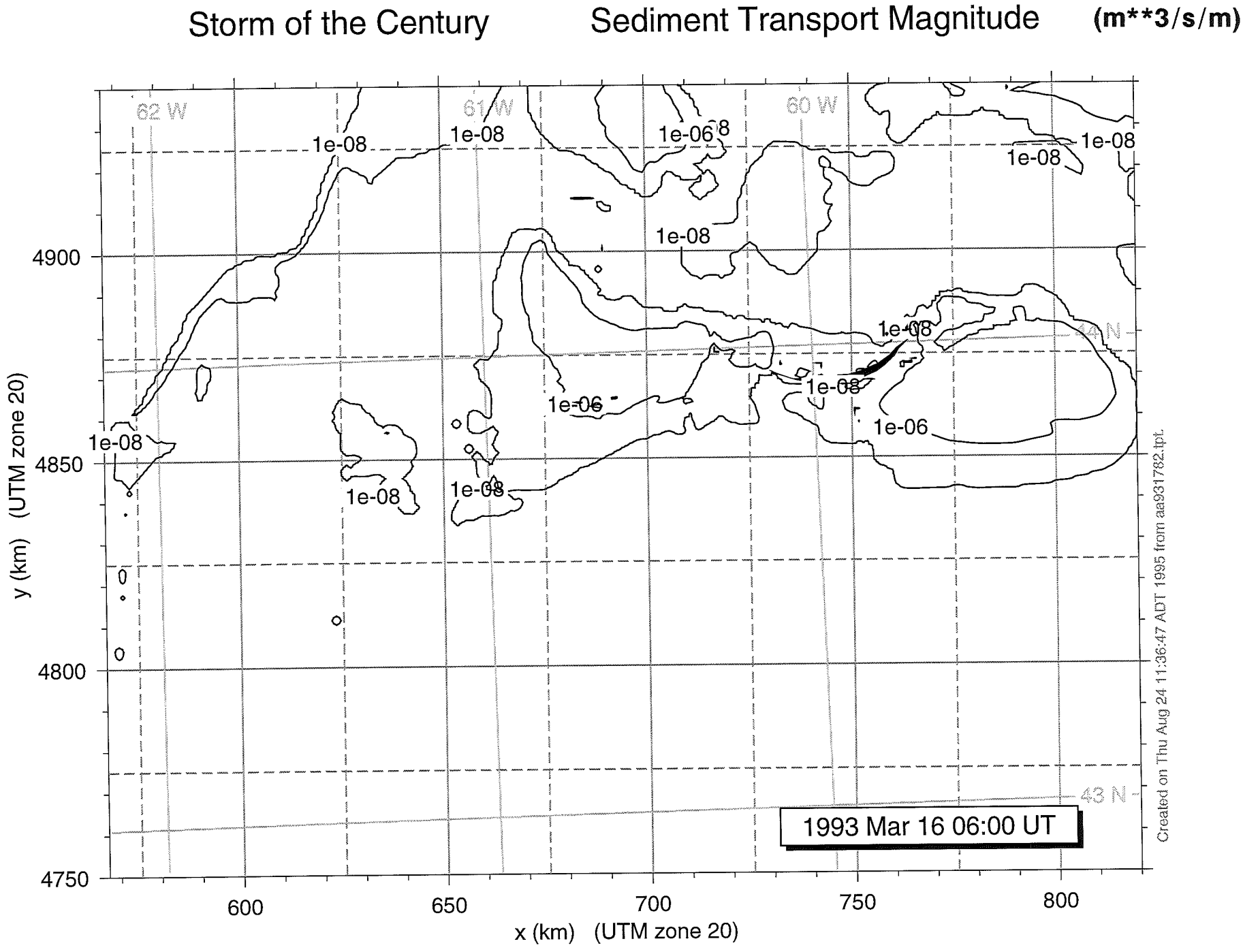

Storm of the Century
Sediment Transport Magnitude
(m**3/s/m)
62 W
61 W
60 W
44 N
43 N
1e-08
1e-06
1993 Mar 16 06:00 UT
4900
4850
4800
4750
600
650
700
750
800
x (km) (UTM zone 20)
y (km) (UTM zone 20)
Created on Thu Aug 24 11:36:47 ADT 1995 from aa931782.tpt.

Figure 7.10

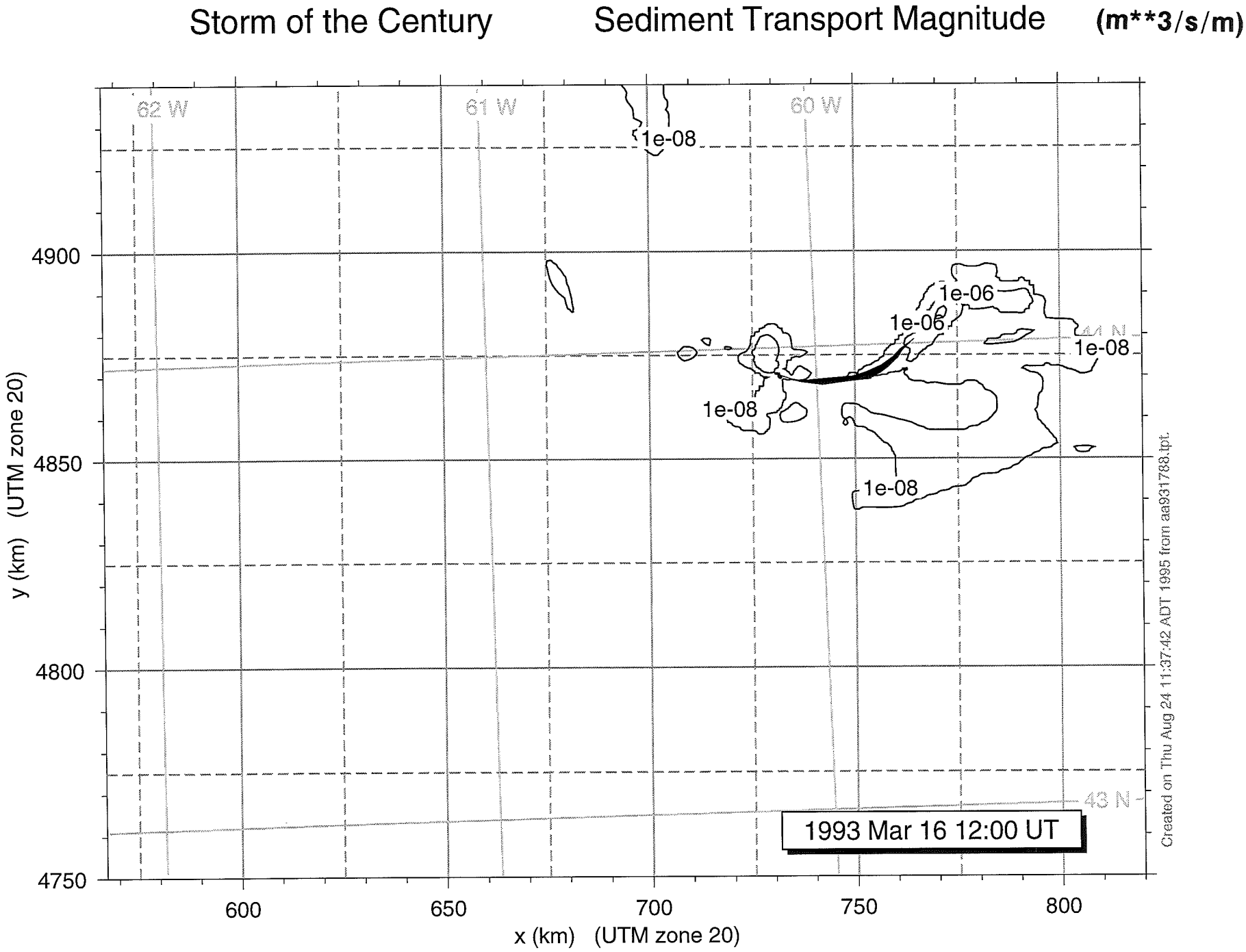

Storm of the Century
Sediment Transport Magnitude
(m**3/s/m)
62 W
61 W
60 W
1e-08
1e-06
1e-06
1e-08
1e-08
1e-08
44 N
43 N
1993 Mar 16 12:00 UT
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
Created on Thu Aug 24 11:37:42 ADT 1995 from aa931788.tpt.

Figure 7.11

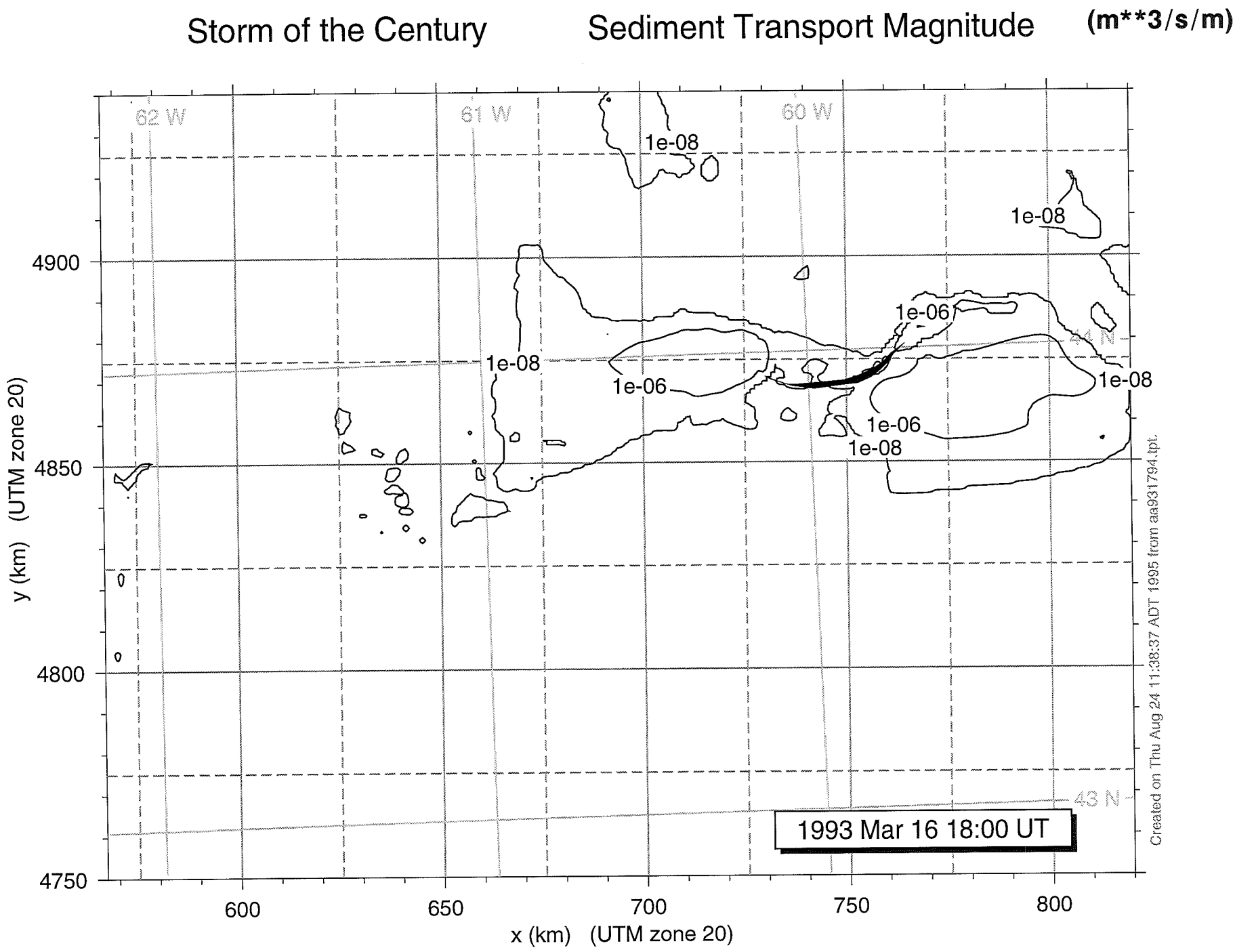

Storm of the Century
Sediment Transport Magnitude
(m**3/s/m)
62 W
61 W
60 W
44 N
43 N
1e-08
1e-06
1993 Mar 16 18:00 UT
4900
4850
4800
4750
600
650
700
750
800
y (km) (UTM zone 20)
x (km) (UTM zone 20)
Created on Thu Aug 24 11:38:37 ADT 1995 from aa931794.tpt.

Figure 7.12

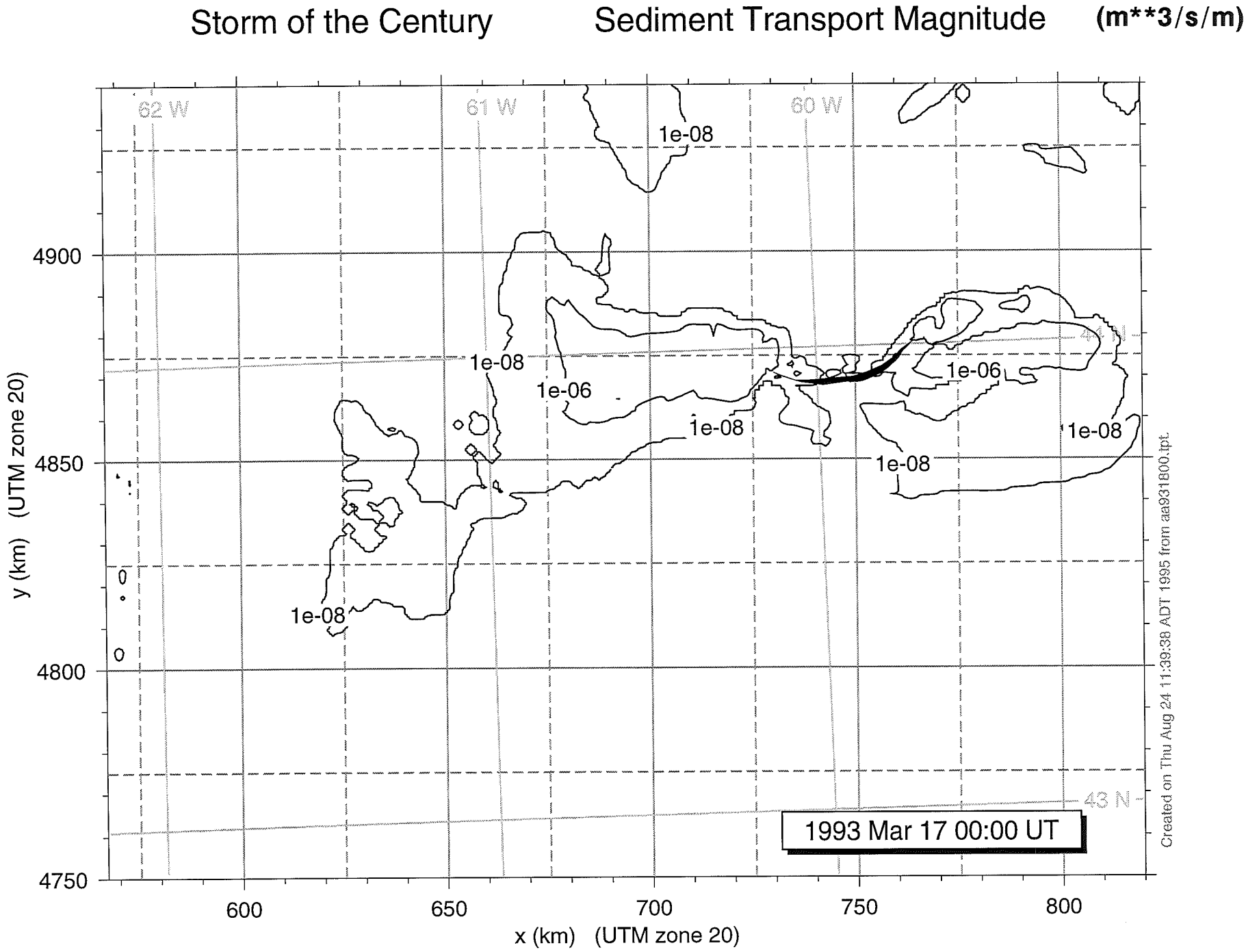

Storm of the Century
Sediment Transport Magnitude
(m**3/s/m)
62 W
61 W
60 W
44 N
43 N
1e-08
1e-06
1e-08
1e-06
1e-08
1e-08
1e-08
1e-08
1993 Mar 17 00:00 UT
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
Created on Thu Aug 24 11:39:38 ADT 1995 from aa931800.tpt

Figure 7.13

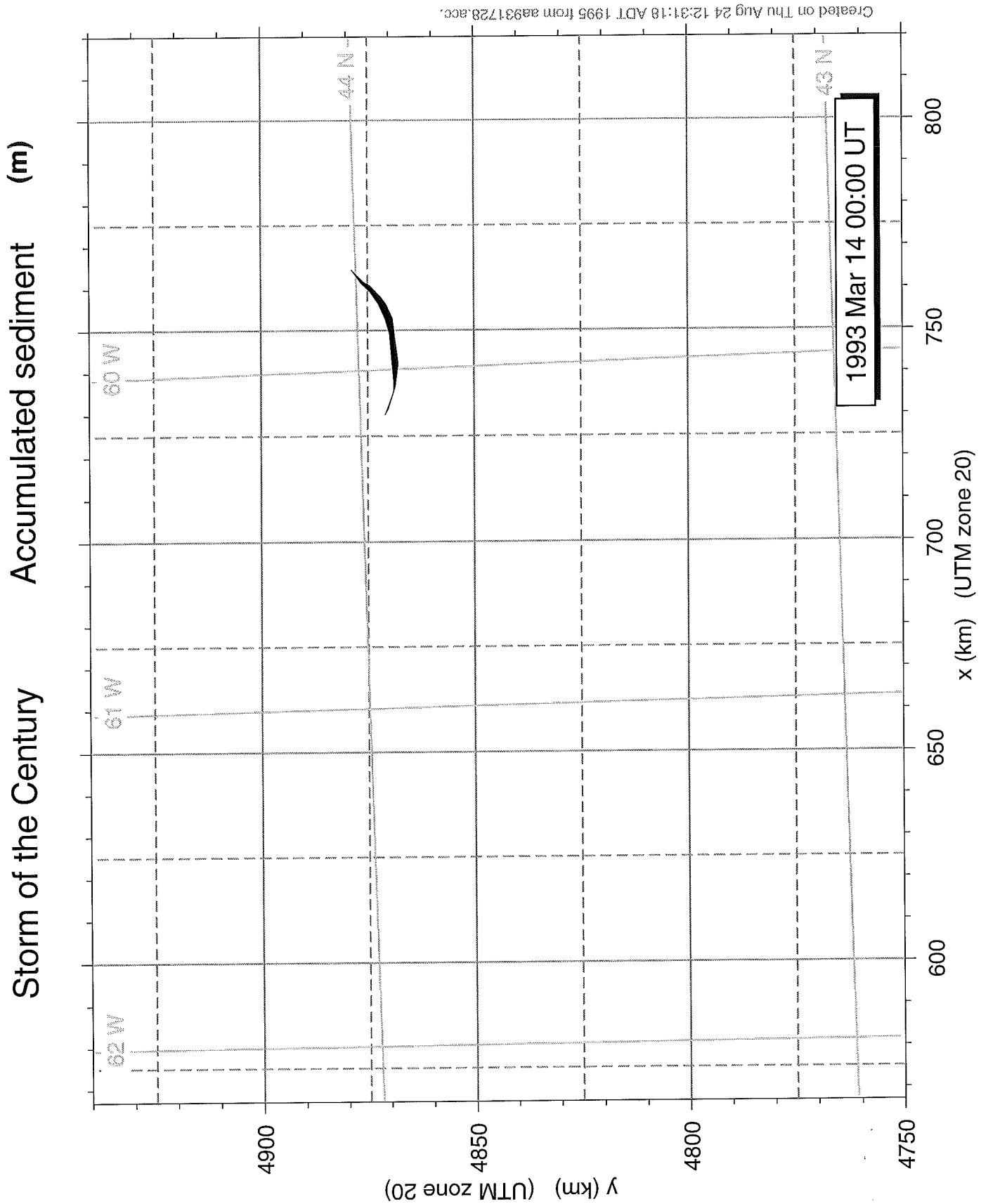

Storm of the Century
Accumulated sediment (m)
Created on Thu Aug 24 12:31:18 ADT 1995 from aa931728.acc.
62 W
61 W
60 W
44 N
43 N
1993 Mar 14 00:00 UT
600
650
700
750
800
x (km) (UTM zone 20)
4900
4850
4800
4750
y (km) (UTM zone 20)

Figure 8.1

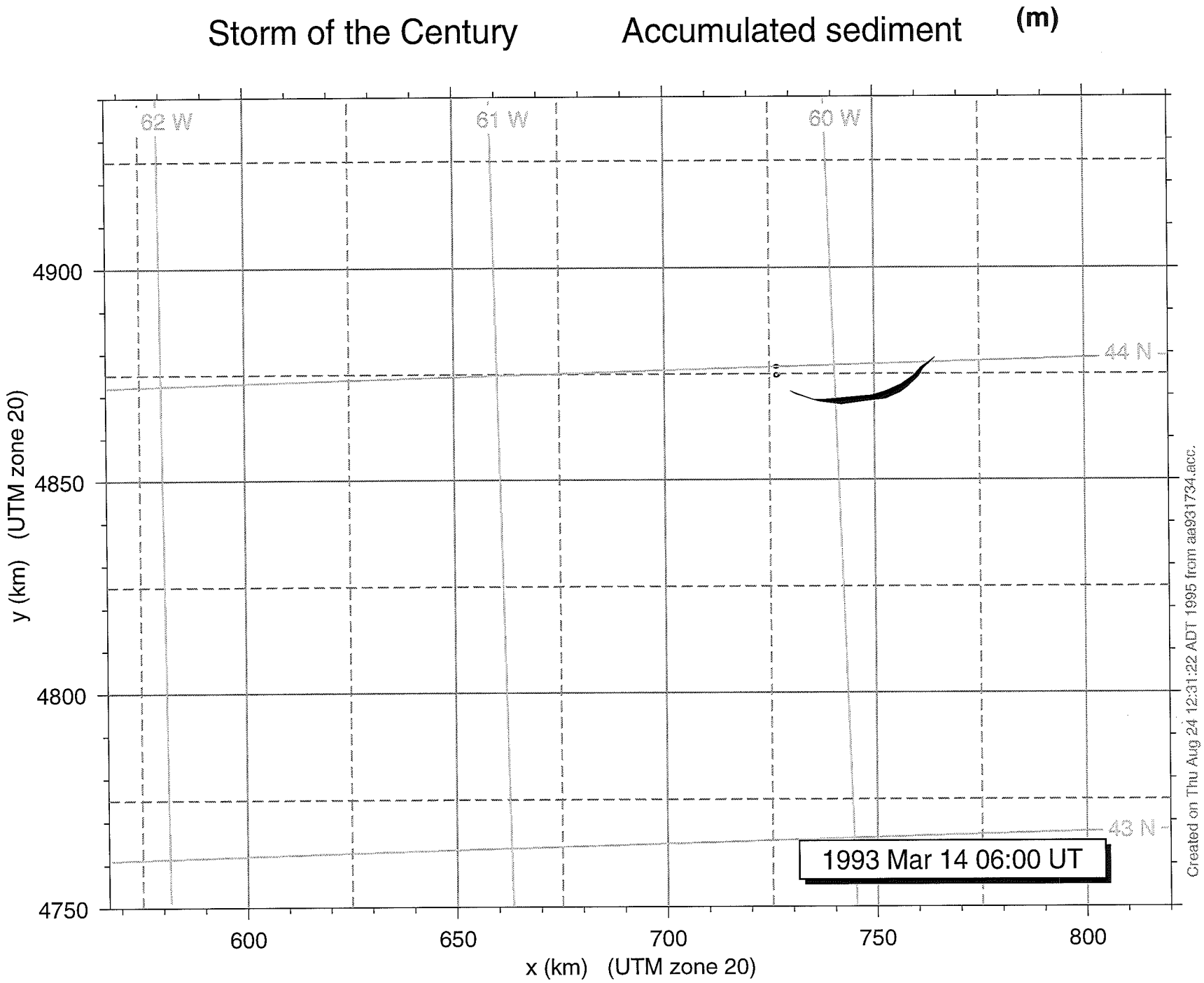

Storm of the Century
Accumulated sediment
(m)
62 W
61 W
60 W
44 N
43 N
1993 Mar 14 06:00 UT
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
Created on Thu Aug 24 12:31:22 ADT 1995 from aa931734.acc.

Figure 8.2

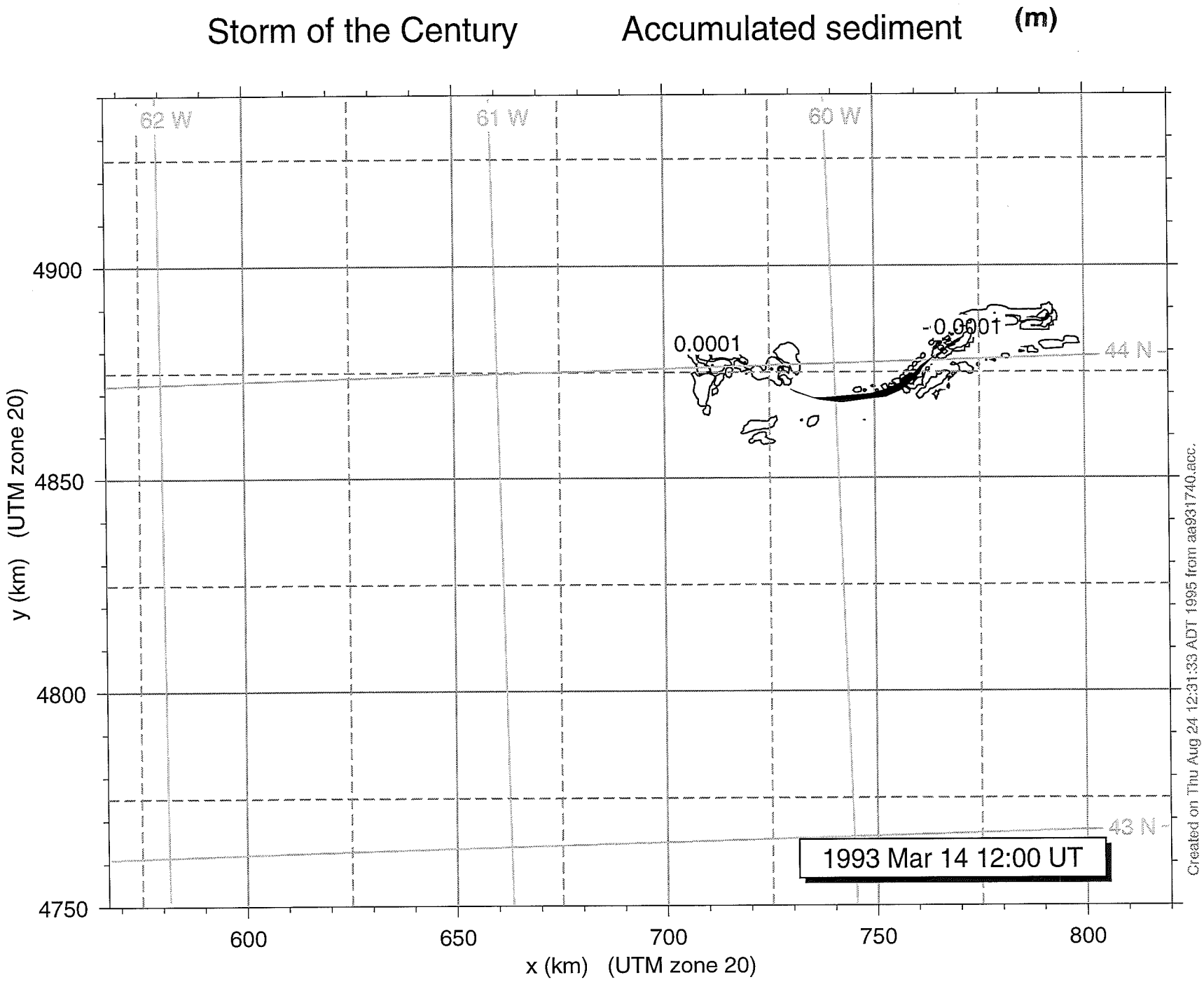

Storm of the Century
Accumulated sediment
(m)
62 W
61 W
60 W
44 N
43 N
0.0001
-0.0001
1993 Mar 14 12:00 UT
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
Created on Thu Aug 24 12:31:33 ADT 1995 from aa931740.acc.

Figure 8.3

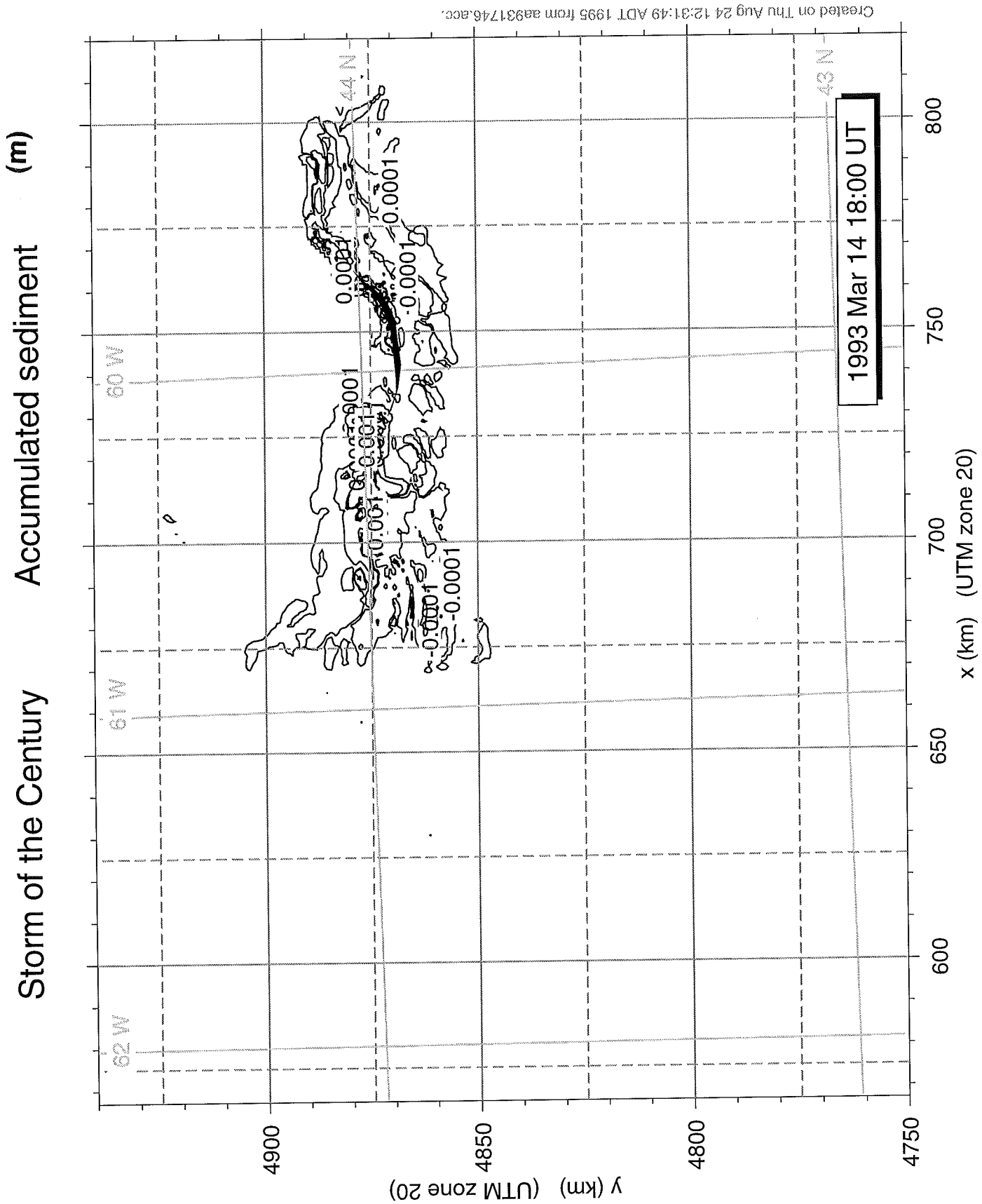

Storm of the Century
Accumulated sediment (m)
1993 Mar 14 18:00 UT
62 W
61 W
60 W
44 N
43 N
0.0001
-0.0001
-0.001
0.001
600
650
700
750
800
x (km) (UTM zone 20)
4900
4850
4800
4750
y (km) (UTM zone 20)
Created on Thu Aug 24 12:31:49 ADT 1995 from aa9317 46.acc.

Figure 8.4

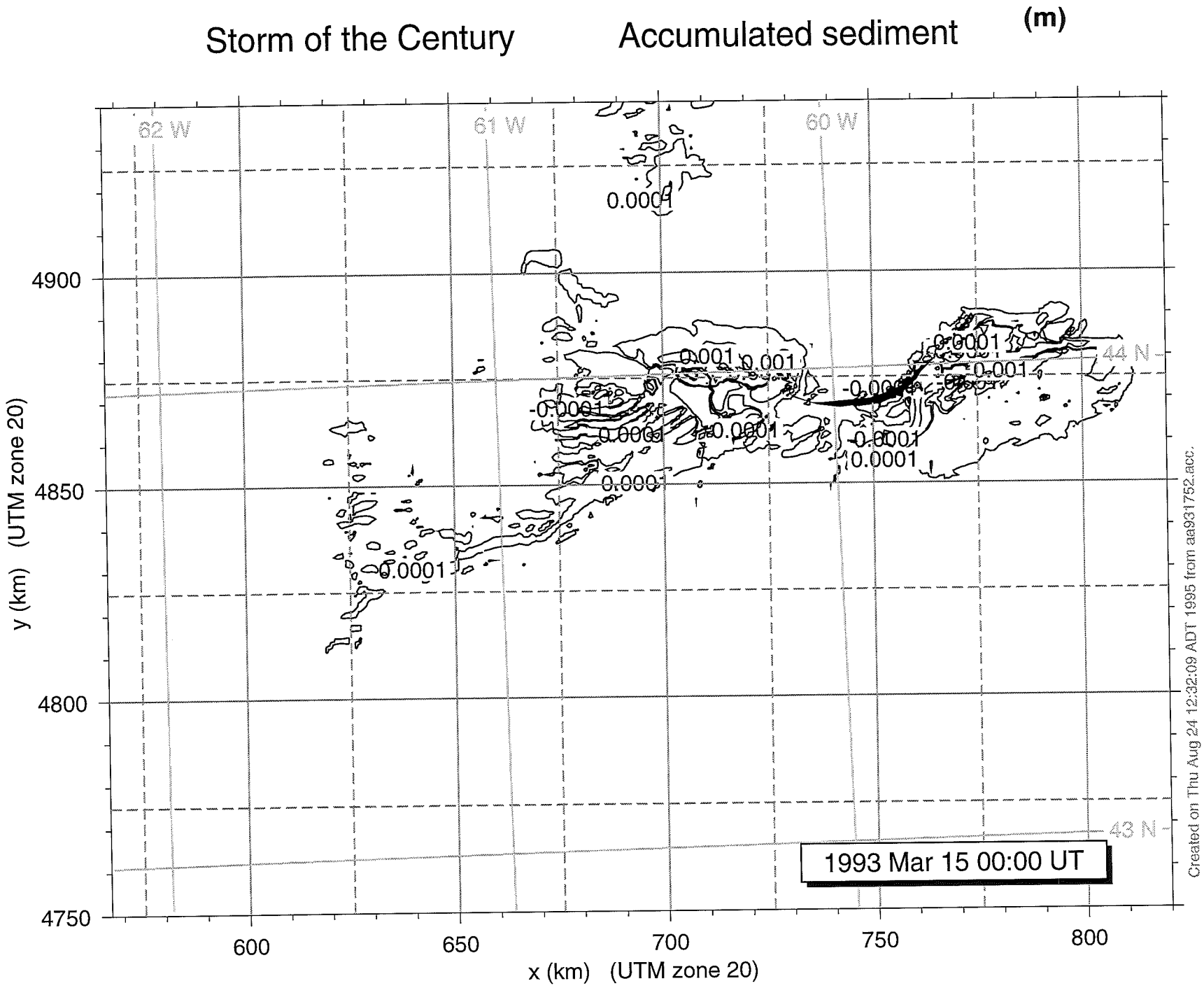

Storm of the Century
Accumulated sediment
(m)
62 W
61 W
60 W
44 N
43 N
0.0001
0.001
-0.0001
1993 Mar 15 00:00 UT
4900
4850
4800
4750
600
650
700
750
800
y (km) (UTM zone 20)
x (km) (UTM zone 20)
Created on Thu Aug 24 12:32:09 ADT 1995 from aa931752.acc.

Figure 8.5

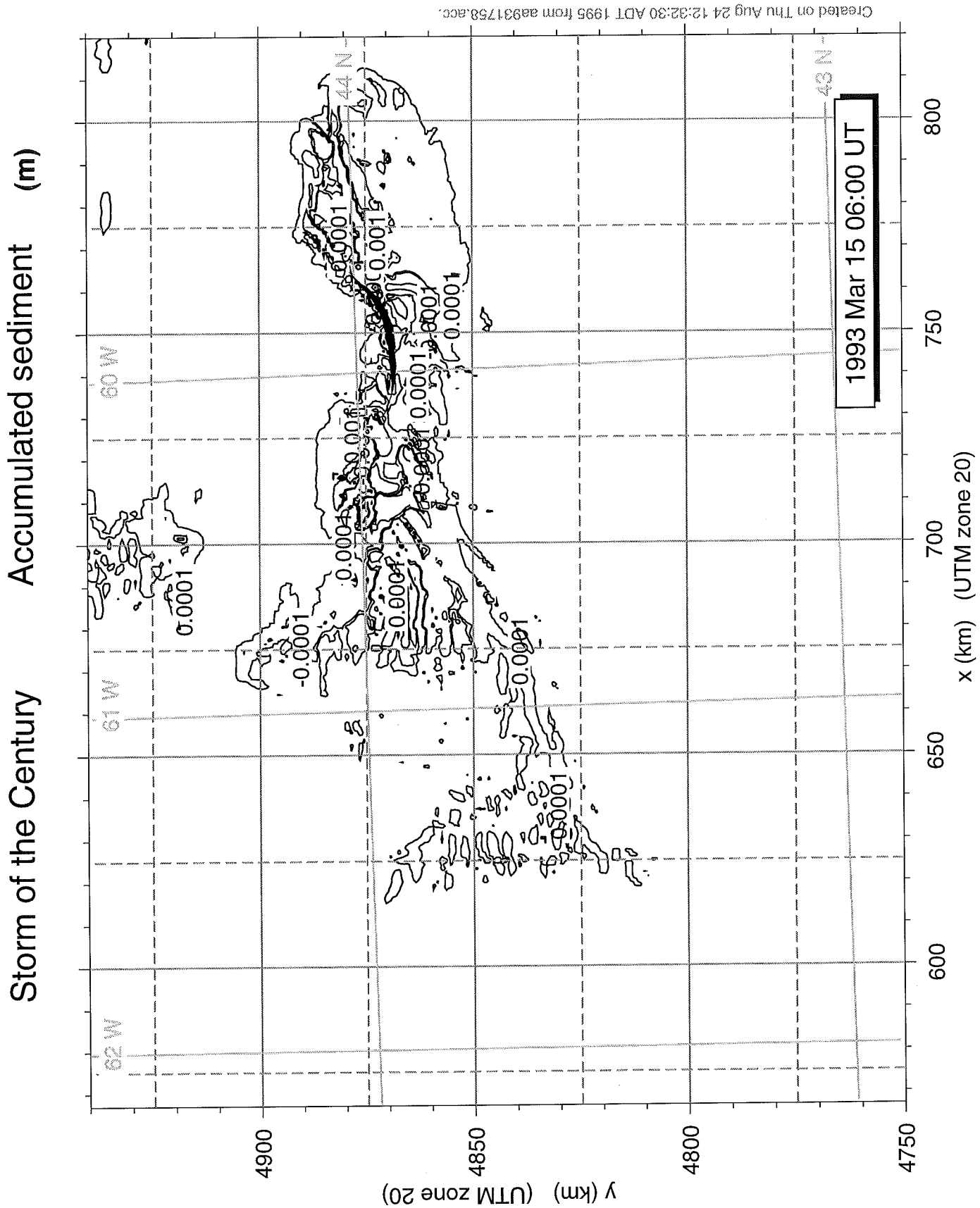

Storm of the Century
Accumulated sediment (m)
Created on Thu Aug 24 12:32:30 ADT 1995 from aa9317 58.acc.
1993 Mar 15 06:00 UT
62 W
61 W
60 W
44 N
43 N
-0.0001
0.0001
0.0015
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)

Figure 8.6

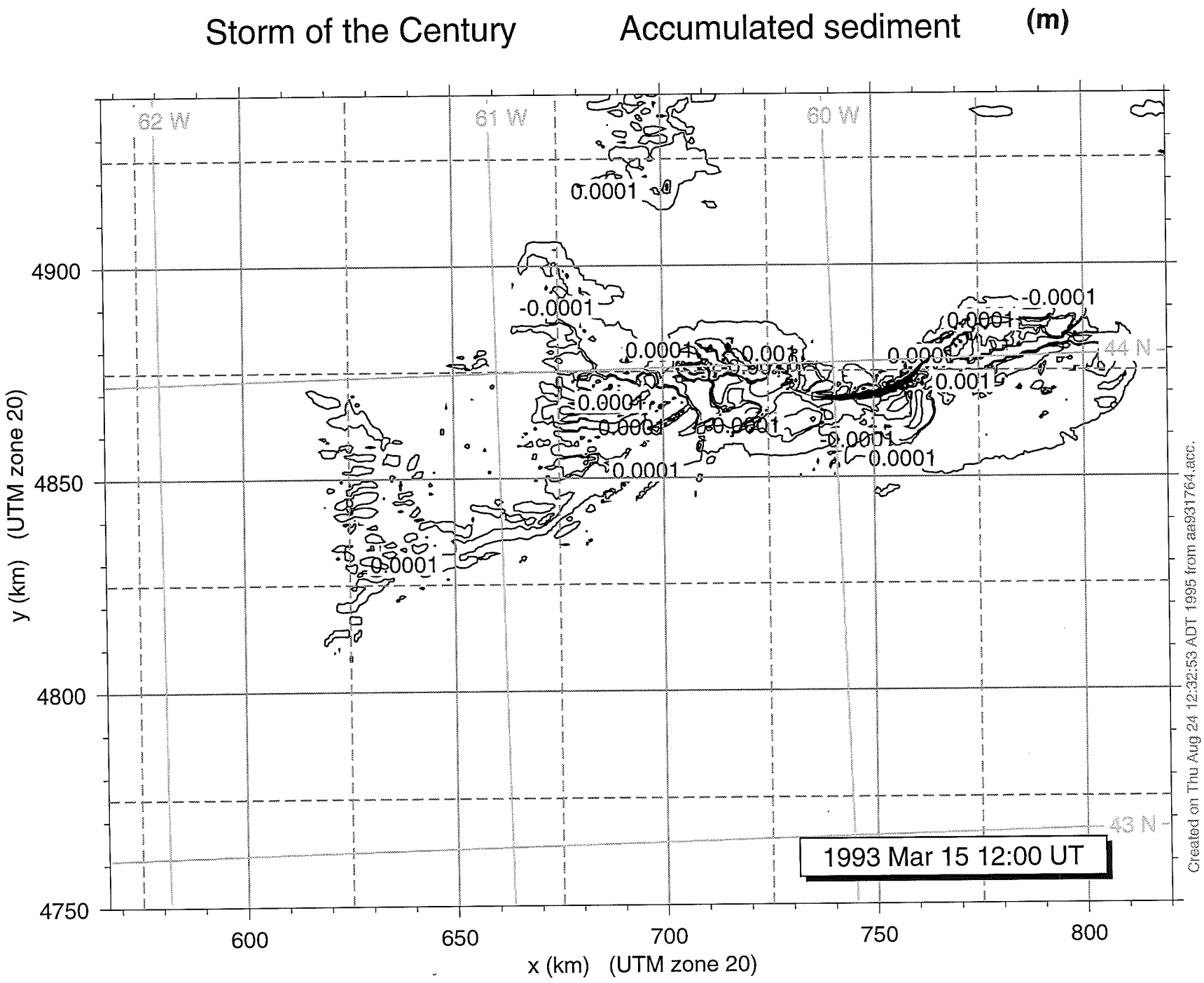

Storm of the Century
Accumulated sediment
(m)
62 W
61 W
60 W
44 N
43 N
0.0001
-0.0001
0.001
1993 Mar 15 12:00 UT
4900
4850
4800
4750
600
650
700
750
800
y (km) (UTM zone 20)
x (km) (UTM zone 20)
Created on Thu Aug 24 12:32:53 ADT 1995 from aa931764.acc.

Figure 8.7

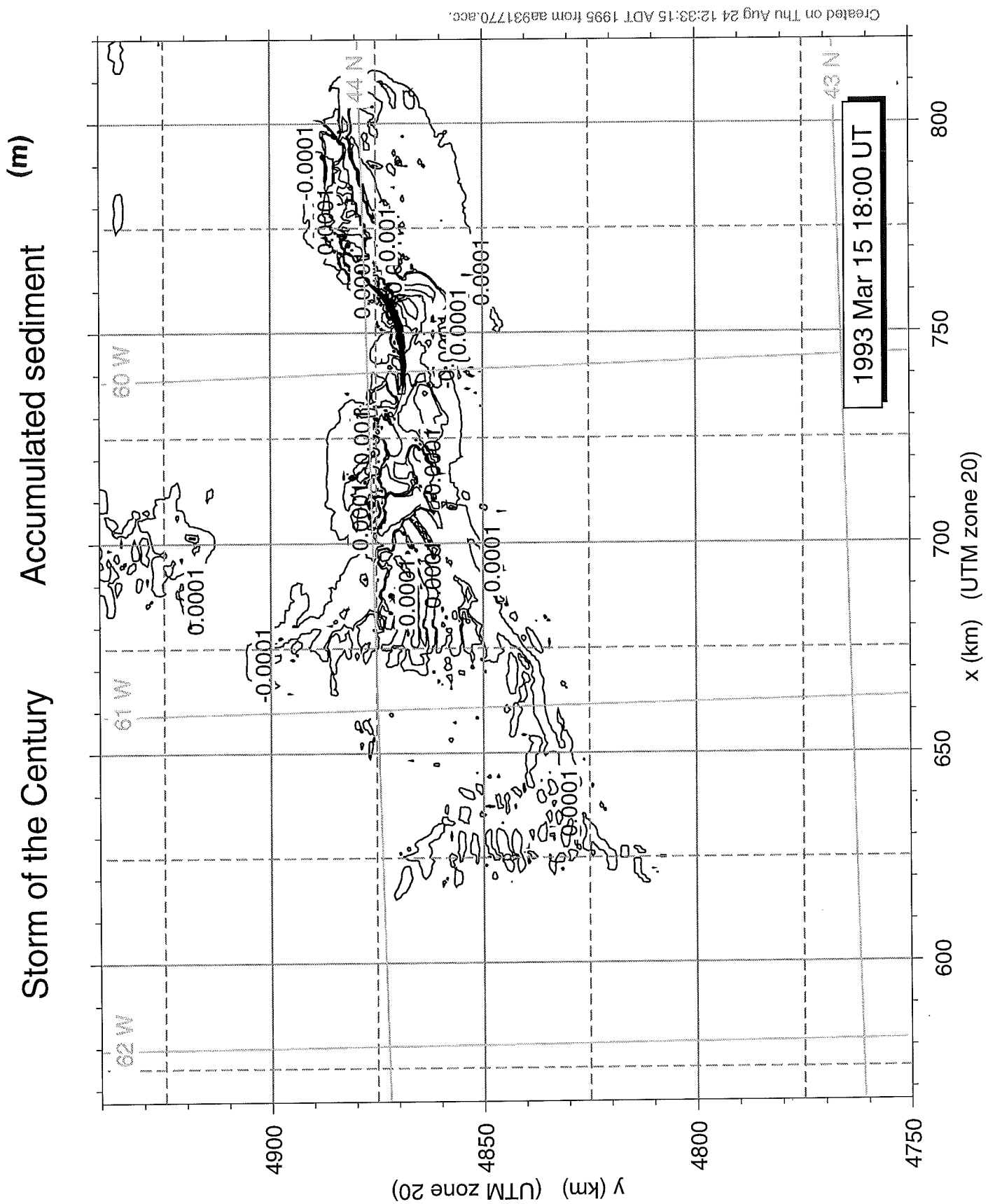

Storm of the Century
Accumulated sediment
(m)
Created on Thu Aug 24 12:33:15 ADT 1995 from aa9317 70.acc.
1993 Mar 15 18:00 UT
62 W
61 W
60 W
44 N
43 N
-0.0001
0.0001
0.001
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)

Figure 8.8

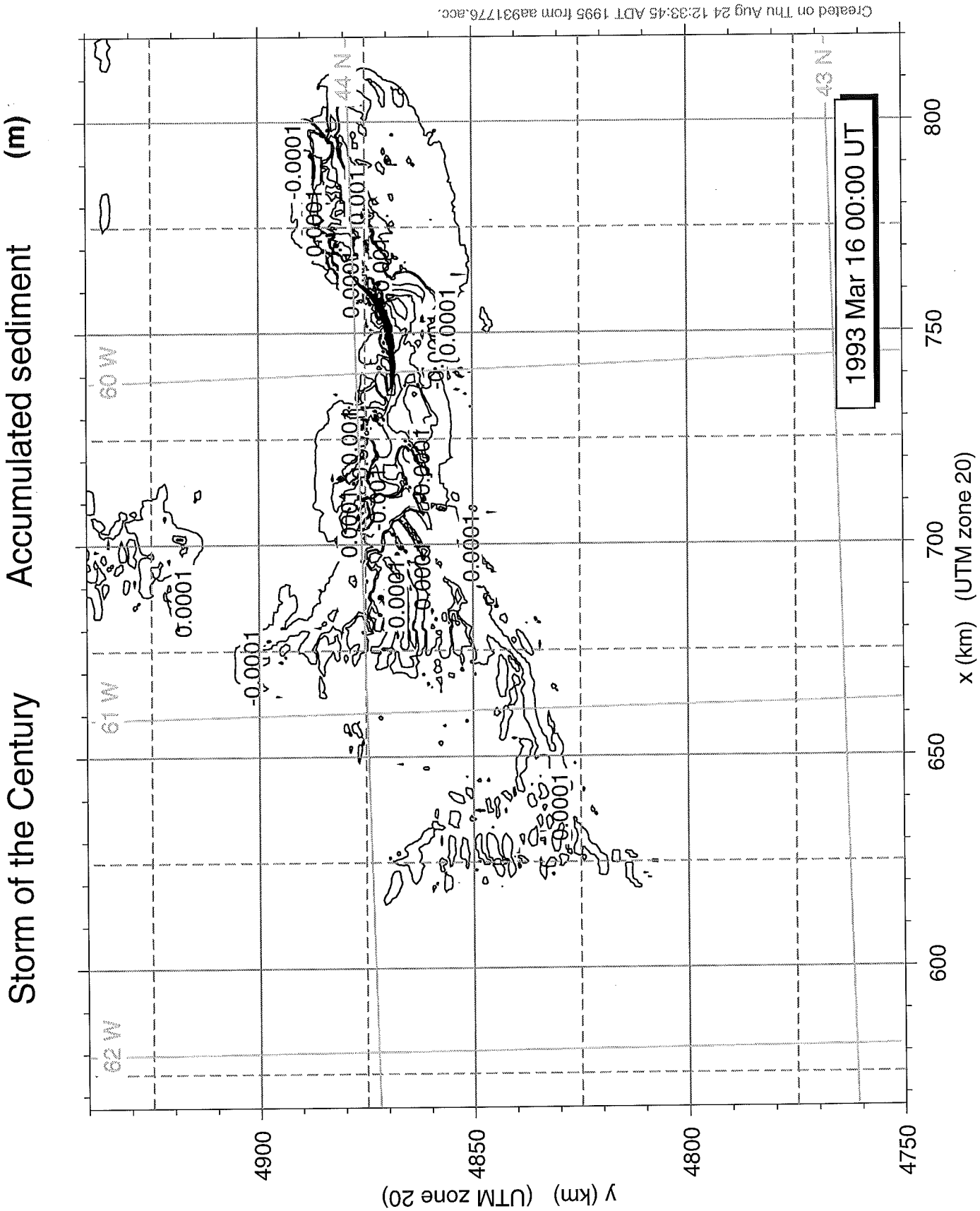

Storm of the Century
Accumulated sediment (m)
1993 Mar 16 00:00 UT
62 W
61 W
60 W
44 N
43 N
-0.0001
0.0001
0.001
0.01
0.1
600
650
700
750
800
x (km) (UTM zone 20)
4900
4850
4800
4750
y (km) (UTM zone 20)
Created on Thu Aug 24 12:33:45 ADT 1995 from aa9317 76.acc.

Figure 8.9

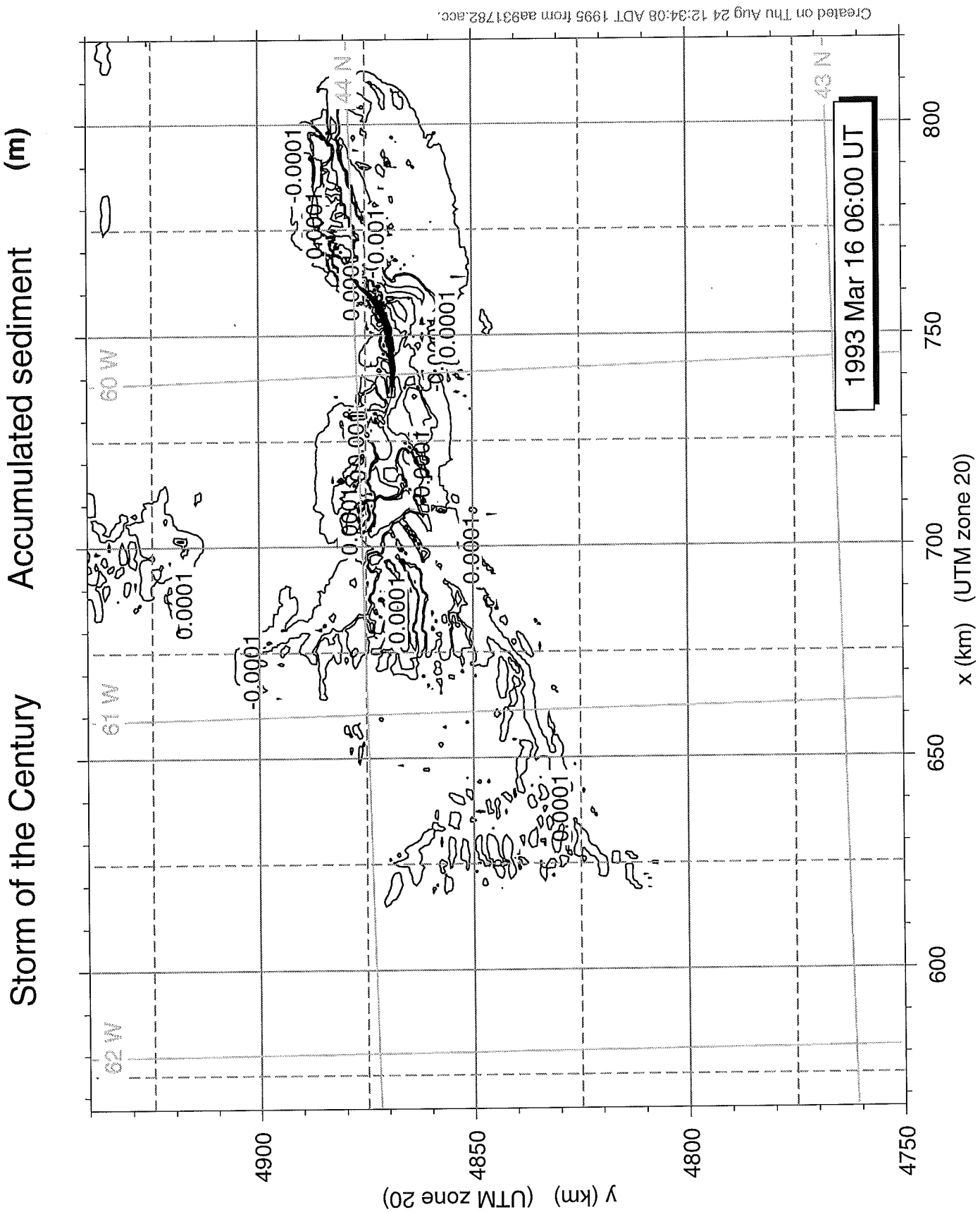

Storm of the Century
Accumulated sediment (m)
1993 Mar 16 06:00 UT
62 W
61 W
60 W
44 N
43 N
-0.0001
0.0001
-0.001
0.001
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
Created on Thu Aug 24 12:34:08 ADT 1995 from aa9317 82.acc.

Figure 8.10

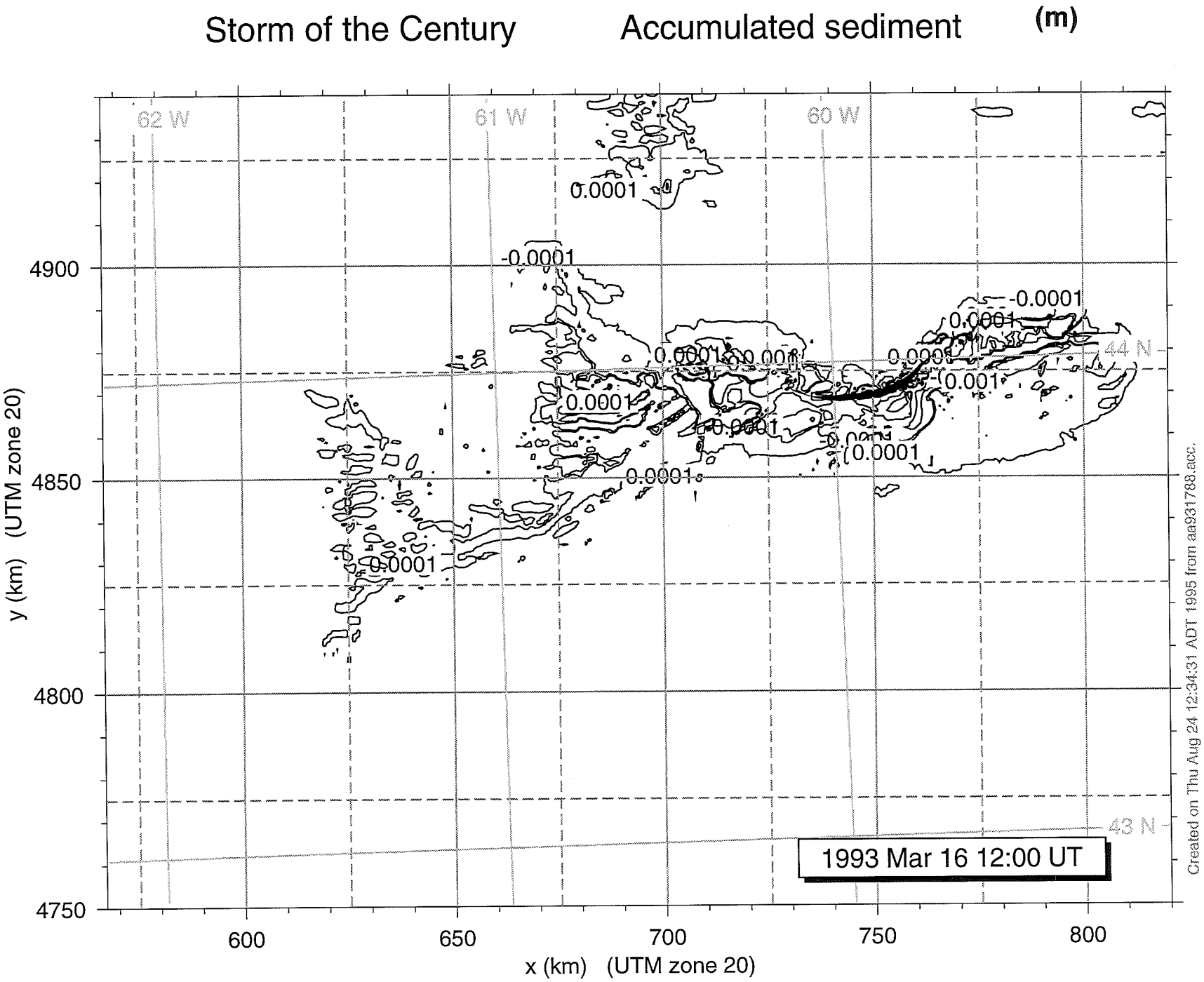

Storm of the Century
Accumulated sediment
(m)
62 W
61 W
60 W
44 N
43 N
0.0001
-0.0001
0.001
1993 Mar 16 12:00 UT
4900
4850
4800
4750
600
650
700
750
800
y (km) (UTM zone 20)
x (km) (UTM zone 20)
Created on Thu Aug 24 12:34:31 ADT 1995 from aa931788.acc.

Figure 8.11

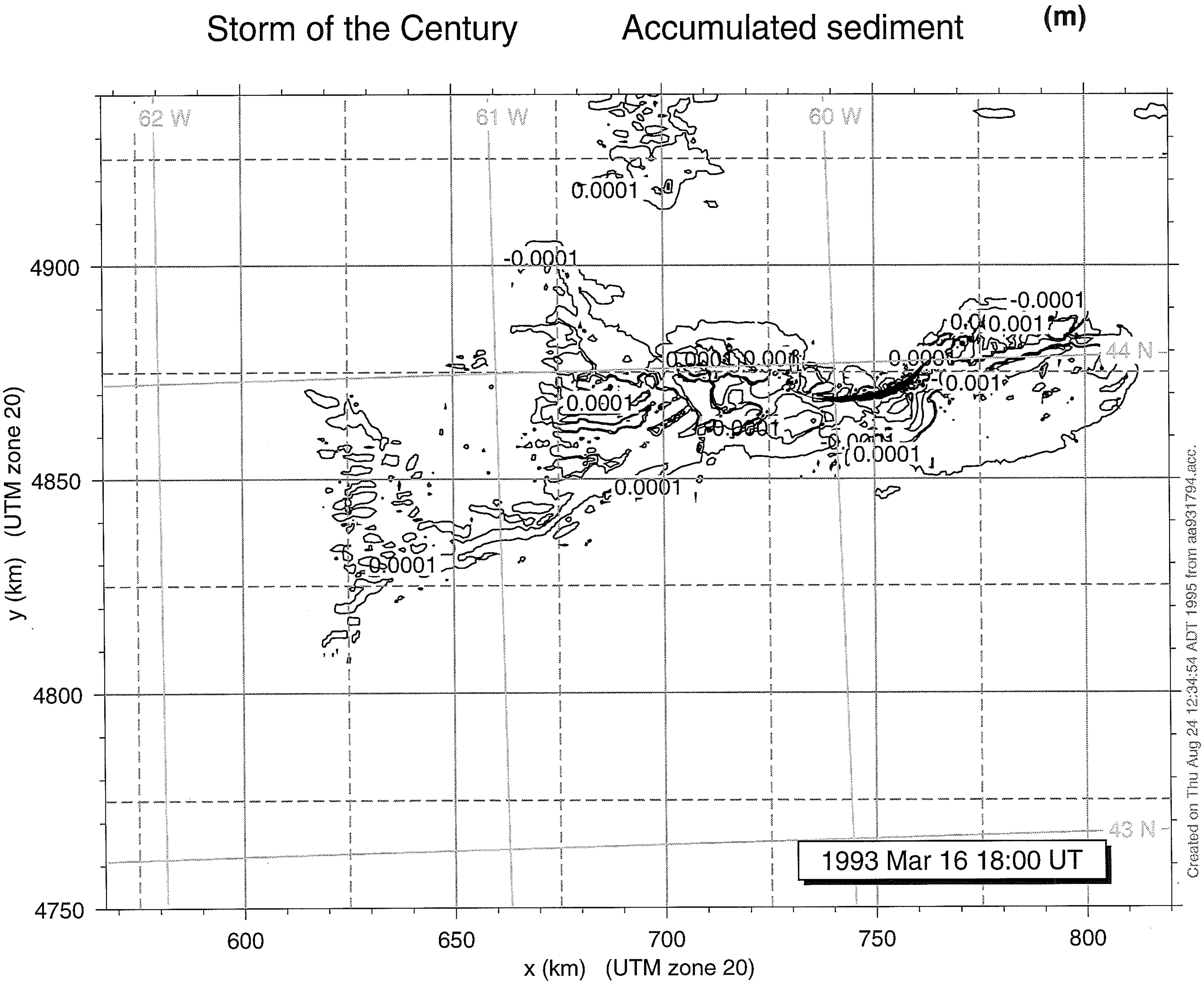

Storm of the Century
Accumulated sediment
(m)
62 W
61 W
60 W
44 N
43 N
0.0001
-0.0001
-0.0001
0.001
-0.001
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
1993 Mar 16 18:00 UT
Created on Thu Aug 24 12:34:54 ADT 1995 from aa931794.acc.

Figure 8.12

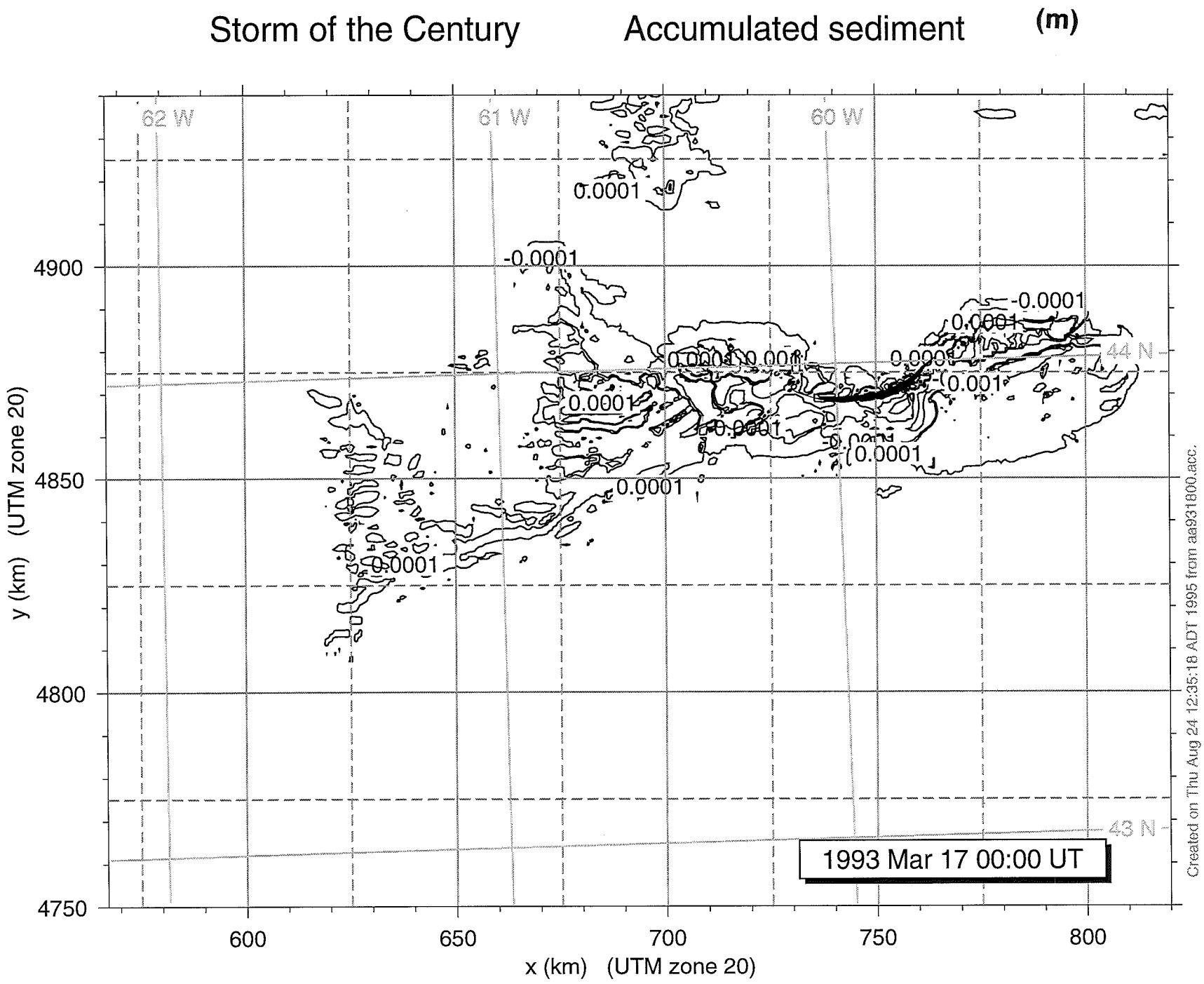

Storm of the Century
Accumulated sediment
(m)
62 W
61 W
60 W
44 N
43 N
0.0001
-0.0001
0.001
-0.001
1993 Mar 17 00:00 UT
4900
4850
4800
4750
y (km) (UTM zone 20)
600
650
700
750
800
x (km) (UTM zone 20)
Created on Thu Aug 24 12:35:18 ADT 1995 from aa931800.acc.

Figure 8.13

APPENDIX A

Flow Charts

Names in single quotes (e.g., 'sed94') are main programs. Other names are subroutines or functions.

1. SETGRID flow chart

```
'setgrid'          Establishes model grid parameters (interactive).
                   (There are no subroutine or function calls.)
```

2. SETUP flow chart

```
'setup'            Establishes parameters for a model run using previously
   |               selected grid.
   |
   |--date2str       Converts year and date to character string time stamp.
   |
   |--date2hrs       Converts date to hour of current year.
   |
   |--names          Composes list of input files required for model run.
```

3. SED94 flow chart

```
'sed94'       Applies sediment transport model SEDTRANS92 over a grid
   |          and through several time steps.
   |
   |--getsetup        Reads model grid and run parameters from [sed94.in].
   |      |
   |      |--putsetup  Writes model run parameters to [sed94.log].
   |      |
   |      |--time      Gets the date from the computer system.
   |      |
   |      |--date      Gets the time from the computer system.
   |
   |--getpar      Reads physical parameters from [sed94.par].
   |
   |--getmap      Reads a map file containing 1, 2, or 3 gridded variables.
   |
   |--setcodes    Sets codes to control calculation of transport divergence.
   |
   |--hrs2date    Converts hour of current year to date.
   |
   |--date2str    Converts year and date to character string time stamp.
   |
   |--hrs2str     Converts year and hour to a character string.
   |
   |--combine     Adds background, tidal, and wind-driven currents
   |
   |--sed92       Computes potential sediment transport
   |      |
   |      |--(See sed92 flow chart on next page.)
   |
   |--divT        Computes divergence of sediment transport.
   |
   |--putmap      Writes a map file containing 1, 2, or 3 REAL variables.
   |
   |--putimap     Writes a map file containing 1, 2, or 3 INTEGER variables.
   |
   |--date        Gets the date from the computer system.
   |
   |--time        Gets the time from the computer system.
```

```
      SUBROUTINE putsetup (runname, idrun,
     1       xmin, ymin, dx, dy, imax, jmax, izone,
     1       startyr, starthr, tstamp, runhrs,
     1       dttide, dtwind, dtwave, dtT, dtS, dtbrk,
     1       deppath, depname, sizpath, sizname, bacpath, bacname,
     1       tidepath, idtide, windpath, idwind, wavepath, idwave)

    IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      integer xmin, ymin, dx, dy
      integer startyr, starthr, runhrs
      integer dttide, dtwind, dtwave, dtT, dtS, dtbrk
      character*2 idrun, idtide, idwind, idwave
      character*8 now
      character*9 today
      character*11 logfile
      character*20 tstamp
      character*40 deppath, sizpath, bacpath, tidepath, windpath, wavepath
      character*20 depname, sizname, bacname
      character*75 runname

* Start log file:
      logfile = idrun//'sed94.log'
      open (unit=21,file=logfile)
      call time (now)
      call date (today)
      write (21, 201) logfile, now, today
      write (21, *) " Program 'sed94'- Sediment transport simulation ",
     1              'using SEDTRANS92.'
      write (21, *)

* Write setup information in log file:
      write (21, 202) runname
      write (21, 204) idrun
      write (21, 208) xmin, ymin, dx, dy, imax, jmax, izone
      write (21, 210) startyr, starthr, tstamp, runhrs
      write (21, 212) dttide, dtwind, dtwave
      write (21, 214) dtT, dtS, dtbrk
      write (21, 220) deppath, depname
      write (21, 222) sizpath, sizname
      write (21, 223) bacpath, bacname
      write (21, 224) tidepath, idtide
      write (21, 226) windpath, idwind
      write (21, 228) wavepath, idwave

      return

201   format (t20, a11, 5x, a8, 5x, a9 /)

202   format (1x, 'Name:' / 1x, "'", a75, "'")
204   format (1x, 'Run ID:' / 1x, "'", a2, "'")
208   format (1x, 'xmin(m.)   ymin(m.)   dx(m)    dy(m)    columns   rows '
     1       '  UTM zone' /
     1       1x, i7, t15, i7, t29, i6, t43, i6, t57, i5, t71, i5, t77, i2)
210   format (1x, 'Startyr  Starthours   Start date/time         Runhours'/
     1       1x, i2, 7x, i4, 9x, "'", a20, "'", 3x, i3)
212   format (1x, 'Input DT (hrs.):  Tidal current   Wind-driven current',
     1       '  Wave field'/
     1       1x, t20, i2, t36, i2, t56, i2)
214   format (1x, 'Output DT (hrs.): Sed. transport  Acc. sed.  Wave breaking'
     1       1x, t20, i2, t36, i2, t42, i2)

220   format (1x, 'Bathymetry map (directory, file name):' /
     1       1x, "'", a40, "'  '", a20"'")
222   format (1x, 'Grain size map (directory, file name):' /
     1       1x, "'", a40, "'  '", a20"'")
223   format (1x, 'Background current map (directory, file name):'/
```

```
     1       1x, "'", a40, "'  '", a20"'")
224   format (1x, 'Tidal current maps (directory, file ID):' /
     1       1x, "'", a40, "'  ", a2)
226   format (1x, 'Wind-driven current maps (directory, file ID):' /
     1       1x, "'", a40, "'  ", a2)
228   format (1x, 'Wave field maps (directory, file ID):' /
     1       1x, "'", a40, "'  ", a2)

      end
```

```
      subroutine getpar (rkb, fract, rhos, rhow, z, iopt1, iopt2,
     1                   conc0, towcd, towce, ws, prs, rkero)

      IMPLICIT DOUBLE PRECISION(A-H,O-Z)

*     'getpar.f' reads physical parameters for a model run from external
*     file 'sed94.par'. The valuse of the parameters are written to the
*     log file previously opened as unit 21.

      open (unit=12,file='sed94.par',status='old')
      read (12, *)
      read (12, *) rkb, fract, rhos, rhow, z, iopt1, iopt2
      read (12, *)
      read (12, *) conc0, towcd, towce, ws, prs, rkero

      close (12)

      write (21, *)
      write (21, *) ' PHYSICAL PARAMETERS:'
      write (21, 202) rkb, fract, rhos, rhow, z, iopt1, iopt2
      write (21, 204) conc0, towcd, towce, ws, prs, rkero

      return

202   format (/ 1x, '    rkb        fract      rhos       rhow       ',
     1              'z          iopt1 iopt2'/
     1       1x, 1p, 5e11.3 , 2i6 )
204   format (/ 1x, '    conc0      towcd      towce      ws         '
     1              'prs        rkero'/
     1       1x, 1p, 6e11.3 /)
      end
```

```
      SUBROUTINE getmap (path, mapfile, imax, jmax, layers,
     1                 array1, array2, array3)

*     'getmap.f' reads variables from external map files into arrays.
*     (Map files can be created by subroutine 'putmap.f'.)

      double precision array1(imax,jmax), array2(imax,jmax), array3(imax,jmax)
      character*60 fullname
      character*40 path
      character*20 mapfile

      write (21, *) '   Get map from ', mapfile
      print *,  '   Get map from ', mapfile
      fullname = path(1:index(path,' ')-1) // mapfile
      open (unit=11,file=fullname,status='old')

      do 10 i = 1, 5
10            read (11, *)

      read (11,*) ((array1(i,j), i = 1, imax), j = jmax, 1, -1)
      if (layers .ge. 2) read (11,*) ((array2(i,j), i=1,imax), j=jmax,1,-1)
      if (layers .eq. 3) read (11,*) ((array3(i,j), i=1,imax), j=jmax,1,-1)

      close (11)


      return

      end
```

```
        subroutine setcodes (depth, imax, jmax, xcode, ycode)

        double precision depth (imax,jmax)
        integer xcode(imax,jmax), ycode(imax,jmax)

*       'setcodes.f' examines the 2-D array 'depth' and assigns action
*       codes to arrays 'xcode' and 'ycode' depending on the environment
*       of each point (i,j). The codes are set as follows:

* xcode  1 = interior cell
*        2 = open boundary W of this cell
*        3 = open boundary E of this cell
*        4 = land W of this cell
*        5 = land E of this cell
*        6 = land cell, no sediment transport calculation
*        7 = deep water cell, no sediment transport calculation

* ycode  1 = interior cell
*        2 = open boundary S of this cell
*        3 = open boundary N of this cell
*        4 = land S of this cell
*        5 = land N of this cell
*        6 = land cell, no sediment transport calculation
*        7 = deep water cell, no sediment transport calculation

* initialize all cells as interior cells:
        do 20 i = 1, imax
                do 20 j = 1, jmax
                        xcode(i,j) = 1
                        ycode(i,j) = 1
20      continue

* set y codes along S and N open boundaries:
        do 30 i = 1, imax
        if (ycode(i,1) .ne. 6) ycode(i,1) = 2           ! open S of this cell
30      if (ycode(i,jmax) .ne. 6) ycode(i,jmax) = 3     ! open N of this cell

* set x codes along W and E open boundaries:
        do 40 j = 1, jmax
        if (xcode(1,j) .ne. 6) xcode(1,j) = 2           ! open W of this cell
40      if (xcode(imax,j) .ne. 6) xcode(imax,j) = 3     ! open E of this cell

* reset codes in all cells that are bounded by land:

        do 50 i = 2, imax-1
                do 50 j = 1, jmax
                if (xcode(i,j) .eq. 6) go to 50
                if (xcode(i-1,j) .eq. 6) xcode(i,j) = 4 ! land W of this cell
                if (xcode(i+1,j) .eq. 6) xcode(i,j) = 5 ! land E of this cell
50      continue

        do 60 i = 1, imax
                do 60 j = 2, jmax-1
                if (ycode(i,j) .eq. 6) go to 60
                if (ycode(i,j-1) .eq. 6) ycode(i,j) = 4 ! land S of this cell
                if (ycode(i,j+1) .eq. 6) ycode(i,j) = 5 ! land N of this cell
60      continue

* reset codes in all land and deep water cells:
        n6 = 0                  ! counter for land cells.
        n7 = 0                  ! counter for deep water points.
        do 70 i = 1, imax
                do 70 j = 1, jmax
*               reset codes in land cells:
                        if (depth(i,j) .le. 0.0) then
                                n6 = n6 + 1
                                xcode(i,j) = 6
                                ycode(i,j) = 6
                        end if
*               reset codes in deep water cells:
                        if (depth(i,j) .ge. 300.0) then
                                n7 = n7 + 1
                                xcode(i,j) = 7
                                ycode(i,j) = 7
                        end if
70      continue

* Record number of interior and land points in 'sed94.out':
        n1 = (imax * jmax) - n6 - n7    ! number of sediment transport cells
        write (21, 201) n1, n6, n7

        return

201     format (/ 1x, 'Grid has ', i6, ' sediment transport cells '/
        1             '          ', i6, ' land cells' /
        1             '          ', i6, ' deep water cells')

        end
```

```
      SUBROUTINE combine (bu, bv, tu, tv, wu, wv, uz, cdir, imax, jmax)

*     'combine' performs vector addition of background current, tidal current,
*     and wind-driven current. Output (uz,cdir) is in (m/s, deg. T).

    IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      dimension bu(imax,jmax), bv(imax,jmax)  ! background current (cm/s)
      dimension tu(imax,jmax), tv(imax,jmax)  ! tidal current (cm/s)
      dimension wu(imax,jmax), wv(imax,jmax)  ! wind-driven current (cm/s)
      dimension uz(imax,jmax), cdir(imax,jmax)! combined current (m/s)

      pi = acos (-1.0d+00)
      rad2deg = 180.0d+00 / pi

      do 5 i = 1, imax
              do 5 j = 1, jmax
                      u = bu(i,j) + tu(i,j) + wu(i,j)          ! (cm/s)
                      v = bv(i,j) + tv(i,j) + wv(i,j)          ! (cm/s)
                      uz(i,j) = sqrt (u * u + v * v) / 100.0   ! (m/s)
                      if (u .eq. 0.0 .and. v .eq. 0.0) then
                              cdir(i,j) = 0.0d+00
                      else
                              cdir(i,j) = 90.0d+00 - (atan2(v,u) * rad2deg)
                              if (cdir(i,j) .lt. 0.0d+00)
     1                                cdir(i,j) = cdir(i,j) + 360.0d+00
                      end if
5     continue

      return

      end
```

```
      SUBROUTINE sed92 (D, GD,                  ! depth, grain size
     1                  UZ, Z, CDIR,            ! ambient current
     1                  HT, PER, WDIR,          ! wave field
     1                  RKB, FRACT, RHOS, RHOW, IOPT1, IOPT2,    ! parameters 1
     1                  CONC0, TOWCD, TOWCE, WS, PRS, RKERO,     ! parameters 2
     1                  Tx,Ty, ibrk)     ! output: T = sediment transport (m**2/s
                                         ! brk = 1.0 if wave is breaking

C***********************************************************************
C
C                SED92: A SEDIMENT TRANSPORT MODEL
C                       FOR CONTINENTAL SHELF CONDITIONS
C
C                       ATLANTIC GEOSCIENCE CENTER-GSC
C                       BEDFORD INSTITUTE OF OCEANOGRAPHY
C                       SEPTEMBER, 1992
C                       Last Modified: January, 1993

C Converted to subroutine 'sed92' by Carl Anderson starting on 25 May 1994,
C under Research and Development contract 23240-4-M026/01-OSC 'A Finite
C Difference Scheme for SEDTRANS92 - The AGC Sediment Transport Model'.

C          Contractor:  Challenger Oceanography
C                       25 Lawnsdale Dr.
C                       Dartmouth, N.S. B3A 2N1
C                       Tel. (902) 469-9756

C Scientific Authority: Dr. Carl L. Amos
C                       Atlantic Geoscience Centre
C                       Bedford Institute of Oceanography
C                       Dartmouth, N.S.

C  THIS PROGRAM CALCULATES SEDIMENT TRANSPORT FOR HORIZONTAL BEDS UNDER
C  COMBINED WAVE AND CURRENT CONDITIONS.  A CHOICE OF TRANSPORT
C  FORMULAE IS AVAILABLE TO THE USER. HOWEVER, NONE OF THESE FORMULAE
C  HAVE BEEN CALIBRATED FOR COMBINED WAVE-CURRENT FLOWS.
C
C  THE AVAILABLE OPTIONS ARE:
C
C  IOPT1 = 1 - ENGELUND-HANSEN (1967) TOTAL LOAD EQUATION
C          2 - EINSTEIN-BROWN (1950) BEDLOAD EQUATION
C          3 - BAGNOLD (1963) TOTAL LOAD EQUATION
C          4 - YALIN (1963) BEDLOAD EQUATION
C          5 - ACKERS-WHITE TOTAL LOAD EQUATION
C          6 - SMITH (1977) SUSPENDED LOAD (WAVES AND CURRENTS)
C          7 - AMOS and GREENBERG (1980) COHESIVE SEDIMENTS
C
C  ALL DIMENSIONAL VARIABLES ARE IN SI UNITS
C***********************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)

        pi = acos (-1.0d+00)
        deg2rad = pi / 180.0d+00
C***********************************************************************

C  CALCULATE WAVE INDUCED BOTTOM VELOCITY AND ORBIT SIZE
C
      CALL OSCIL(HT,PER,D,UB,AB,WL,ibrk)
        if (ibrk .eq. 1) then
                Tx = 0.0
                Ty = 0.0
                return
        end if
C
C***********************************************************************
C CALCULATE FRICTION FACTOR, AMBIENT CURRENT AND BOTTOM STRESSES.
```

```
C
      CALL FRICFAC(UZ,Z,CDIR,UB,AB,PER,WDIR,GD,RKB,RHOW,IOPT2,RKBC,
     1FCW,UA,PHIB,U100,PHI100,USTCSGM,USTWSGM,USTCWSGM,USTC,USTW,
     1USTCW)
C
C***********************************************************************
C CALCULATE THRESHOLD CRITERIA FOR SEDIMENT TRANSPORT
C
C   NOT APPLICABLE FOR COHESIVE SEDIMENTS THEREFORE SKIP IF NECESSARY
        IF (IOPT1.EQ.7) GOTO 1100
C
      CALL THRESH(D,UB,GD,RHOS,RHOW,IOPT1,FCW,USTCRB,USTCRS,
     @VCB,VCS,DGR,AGR,RGR)

C***********************************************************************
C  CALCULATE THE DURATION OF THE DIFFERENT SEDIMENT TRANSPORT PHASES
C
      CALL TIMING(UA,PHIB,UB,PER,U100,USTCRB,USTCRS,USTCSGM,
     @USTWSGM,USTCWSGM,TB1,TB2,TS1,TS2,PERBED,PERSUSP,
     @USTCS1,USTCS,USTWS,USTCWS,USTCWSM)
C
 1100    CONTINUE
C***********************************************************************
C CALCULATE SEDIMENT TRANSPORT RATE AND DIRECTION
C
      CALL TRANSPO(D,UA,UB,U100,PHIB,PHI100,FCW,PER,GD,RKB,FRACT,
     @RHOS,RHOW,VCB,VCS,USTCRB,USTCS,USTWS,USTCWS,AGR,DGR,CDIR,WDIR,
     @TB1,TB2,TS1,TS2,PERBED,PERSUSP,IOPT1,IOPT2,CONC0,TOWCD,TOWCE,
     @WS,PRS,RKERO,Tq,SEDM,Tdir,SEDXC,SEDXT,SEDY,CONC)

C Compute x and y components of sediment transport:
        angle = (90.0d+00 - Tdir) * deg2rad
        Tx = Tq * cos (angle)
        Ty = Tq * sin (angle)

C***********************************************************************

        return
      END
```

```
      SUBROUTINE divT (i, j, imax, jmax, dx, dy, Tx, Ty,
     1                        dTxdx, dTydy, xcode, ycode)

*     Calculates horizontal divergence at (i,j) of vector function T whose
*     x and y vector components are contained in matrices Tx and Ty.
*     Specifically, calculates d(Tx)/dx and d(ty)/dy.
*     Xcode and ycode determine the finite difference formulas to be used
*     depending on the environment of the location (i,j) in question.

    IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      integer xcode(imax,jmax), ycode(imax,jmax), dx, dy
      dimension Tx(imax,jmax), Ty(imax, jmax)

* xcode:        1 = interior cell
*               2 = open boundary W of this cell
*               3 = open boundary E of this cell
*               4 = land W of this cell
*               5 = land E of this cell
*               6 = land cell
*               7 = deep water cell

      go to (11,12,13,14,15,16,17) xcode(i,j)

11    dTxdx =  (Tx(i+1,j) - Tx(i-1,j)) / 2.0 / dx
               go to 20
12    dTxdx =  (Tx(i+1,j) - Tx(i,j)) / dx
               go to 20
13    dTxdx =  (Tx(i,j) - Tx(i-1,j)) / dx
               go to 20
14    dTxdx =  (Tx(i,j) + Tx(i+1,j)) / 2.0 / dx
               go to 20
15    dTxdx = -(Tx(i,j) + Tx(i-1,j)) / 2.0 / dx
               go to 20
16    dTxdx =  0.0
               go to 20
17    dTxdx =  0.0

* ycode:        1 = interior cell
*               2 = open boundary S of this cell
*               3 = open boundary N of this cell
*               4 = land S of this cell
*               5 = land N of this cell
*               6 = land cell
*               7 = deep water cell

20    go to (21,22,23,24,25,26,27) ycode(i,j)

21    dTydy =  (Ty(i,j+1) - Ty(i,j-1)) / 2.0 / dy
               return
22    dTydy =  (Ty(i,j+1) - Ty(i,j)) / dy
               return
23    dTydy =  (Ty(i,j) - Ty(i,j-1)) / dy
               return
24    dTydy =  (Ty(i,j) + Ty(i,j+1)) / 2.0 / dy
               return
25    dTydy = -(Ty(i,j) + Ty(i,j-1)) / 2.0 / dy
               return
26    dTydy =  0.0
               return
27    dTydy =  0.0
               return

      end
```

```
      SUBROUTINE putimap (mapfile, title1, title2,
     1                xmin, ymin, dx, dy, imax, jmax, izone,
     1                year, yrhour, tstamp, layers, array1, array2, array3)

*     'putmap.f' writes INTEGER variables from INTEGER arrays to external file
*     (Map files can be read by subroutine 'getmap.f'.)

    IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      integer xmin, ymin, dx, dy
      integer year, yrhour
      character*3 n3, cols
      character*20 tstamp
      character*20 mapfile
      character*30 title1, title2
      integer array1(imax,jmax), array2(imax,jmax), array3(imax,jmax)

      write (21, *) '   Put map in ', mapfile
      print *, '   Put map in ', mapfile

      write (n3, '(i3)') imax
      read  (n3, '(a3)') cols

      open (unit=22,file=mapfile)
      write (22, '(a30)') title1
      write (22, '(a30)') title2
      write (22,  *) xmin, ymin, dx, dy, imax, jmax, layers, izone
      write (22,  *) year, yrhour
      write (22,  '(a20)') tstamp

      write (22, '(1x, ' //cols// 'i5)')
     1       ((array1(i,j), i = 1, imax), j = jmax, 1, -1)
      if (layers .ge. 2) write (22, '(1x, ' //cols// 'i5)')
     1       ((array2(i,j), i = 1, imax), j = jmax, 1, -1)
      if (layers .eq. 3) write (22, '(1x, ' //cols// 'i5)')
     1       ((array3(i,j), i = 1, imax), j = jmax, 1, -1)
      close (22)

      return

      end
```

```
      SUBROUTINE putmap (mapfile, title1, title2,
     1               xmin, ymin, dx, dy, imax, jmax, izone,
     1               year, yrhour, tstamp, layers, array1, array2, array3)

*     'putmap.f' writes variables from arrays to external files.
*     (Map files can be read by subroutine 'getmap.f'.)

    IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      integer xmin, ymin, dx, dy
      integer year, yrhour
      character*3 n3, cols
      character*20 tstamp
      character*20 mapfile
      character*30 title1, title2
      dimension array1(imax,jmax), array2(imax,jmax), array3(imax,jmax)

      write (21, *) '   Put map in ', mapfile
      print *, '   Put map in ', mapfile

      write (n3, '(i3)') imax
      read  (n3, '(a3)') cols

      open (unit=22,file=mapfile)
      write (22, '(a30)') title1
      write (22, '(a30)') title2
      write (22,  *) xmin, ymin, dx, dy, imax, jmax, layers, izone
      write (22,  *) year, yrhour
      write (22,  '(a20)') tstamp

      write (22, '(1x, 1p, ' //cols// 'e12.4)')
     1       ((array1(i,j), i = 1, imax), j = jmax, 1, -1)
      if (layers .ge. 2) write (22, '(1x, 1p, ' //cols// 'e12.4)')
     1       ((array2(i,j), i = 1, imax), j = jmax, 1, -1)
      if (layers .eq. 3) write (22, '(1x, 1p, ' //cols// 'e12.4)')
     1       ((array3(i,j), i = 1, imax), j = jmax, 1, -1)
      close (22)

      return

      end
```

```
      integer function date2hrs (year, month, day, hour)

*     Converts date and time to hours from 00:00 hrs 1 Jan. of 'year'.

      integer year, day, hour, days(11), hours
      data days /31, 59,  90, 120, 151, 181, 212, 243, 273, 304, 334/

      leap = 0
      if (mod (year, 4) .eq. 0) leap = 24

      if (month .eq. 1) hours = (day - 1) * 24 + hour
      if (month .eq. 2) hours = 744 + (day - 1) * 24 + hour
      if (month .ge. 3) hours = days(month-1) * 24 + leap + (day-1)*24 + hour
      date2hrs = hours

      return
      end
      character*20 FUNCTION date2str (year, month, day, hour)

*     Creates date/time stamp character string like '1994 Jul 20 18:00 UT'

      character*3 mstring(12)
      character*2 n2(3), ystring, dstring, hstring
      integer year, day, hour
      data mstring / 'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
     1               'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec' /

      write (n2, '(i2.2)') year, day, hour
      read  (n2, '(a2)') ystring, dstring, hstring

      date2str = '19'//ystring//' '//mstring(month)//' '//dstring
     1       //' '//hstring//':00 UT'

      return
      end
      SUBROUTINE hrs2date (yrhour, year, month, day, hour)
* converts hours since 00:00 1 Jan. to date and time.

      integer yrhour, year, month, day, hour, days(12)
      data days /31, 59, 90, 120, 151, 181, 212, 243, 273, 304,
     1       334, 365/

      yr = mod (year, 1900)
      leap = 0
      if (mod (yr, 4) .eq. 0) leap = 24
      if (yrhour .gt. 8759 + leap) then
              yrhour = yrhour - 8760 - leap
              yr = yr + 1
              leap = 0
              if (mod (yr, 4) .eq. 0) leap = 24
      end if

      do 10 month = 1, 12
10            if (yrhour .lt. days(month) * 24 + leap) go to 11
11    continue

      if (month .eq. 1) day = yrhour / 24 + 1
      if (month .eq. 2) day = (yrhour - 744) / 24 + 1
      if (month .ge. 3) day = (yrhour -days(month-1)*24 -leap)/24 + 1
      hour = mod (yrhour, 24)

      return
      end
      character*7 FUNCTION hrs2str (year, yearhr)

*     'hrs2str.f' converts 'year' and 'yearhr' (hours from 00:00 hrs 1 Jan.
*     of 'year') to a character string 'yyhhhh.'.

      character*2 n2, yy
      character*4 n4, hhhh
      integer year, yearhr

      write (n2, '(i2)') year
      read  (n2, '(a2)') yy
      write (n4, '(i4.4)') yearhr
      read  (n4, '(a4)') hhhh
      hrs2str = yy // hhhh // '.'

      return
      end
```

```
      PROGRAM setgrid ! sets up grid for sediment transport model sed94.f

      integer xmin, ymin, xmax, ymax, dx, dy, xspan, yspan

      print *
      print *, 'This version of "sed94" uses a rectangular grid in the'
      print *, 'Universal Transverse Mercator (UTM) coordinate system.'
      print *

      print *, '  Enter UTM zone: '
      read *, izone

      print *, '  Enter the x-coordinate of the SW corner of the grid (in m.):
      read *, xmin

      print *, '  and the y-coordinate of the SW corner (in m.): '
      read *, ymin

      print *, '  Enter the x-coordinate of the NE corner (in m.): '
      read *, xmax

      print *, '  and the y-coordinate of the NE corner (in m.): '
      read *, ymax

      print *, " Enter the grid resolution (dx = dy) in meters: "
      read *, dx
      dy = dx

* Compute number of columns (imax), rows (jmax), and total points (npts):
      ncols = (xmax - xmin) / dx
      nrows = (ymax - ymin) / dy
      npts = ncols * nrows

* Compute dimensions of domain in km:
      xspan = ncols * dx / 1000
      yspan = nrows * dy / 1000

      print *
      print *
      print *, 'The grid you have specified:'
      print *
      print *, 'Grid coordinates refer to UTM grid zone ', izone, '.'
      print *
      print *, 'SW corner of grid is at x = ', xmin, ' m, y = ', ymin, ' m.'
      print *, 'NE corner of grid is at x = ', xmax, ' m, y = ', ymax, ' m.'
      print *
      print *, 'W-E grid dimension = ', xspan, ' km.'
      print *, 'S-N grid dimension = ', yspan, ' km.'
      print *
      print *, 'Grid resolution   = ', dx, ' m.'
      print *
      print *, 'Number of columns = ', ncols
      print *, 'Number of rows    = ', nrows

* Write grid information to output file:
      open (unit=21,file='setgrid.out')

      write (21, '(2i10, 5i6)') xmin, ymin, dx, dy, ncols, nrows, izone
      close (21)

      stop

      end
```

```
      PROGRAM setup   ! creates control file 'sed94.in' and the include
                      ! file 'sed94.inc' for program 'sed94.f'

      character*2 idrun, idtide, idwind, idwave
      character*20 date2str, tstamp
      character*40 deppath, sizpath, bacpath
      character*40 tidepath, windpath, wavepath, path
      character*20 depname, sizname, bacname
      character*75 runname
      integer xmin, ymin, dx, dy, day, hour, startyr, starthr, date2hrs
      integer runhrs, dttide, dtwind, dtwave, dtT, dtS, dtbrk
      data dttide / 1 /

* Prompt user for 2-character run ID:
      print *, 'Enter 2-character IDENTIFIER for this model run: '
      read *, idrun

* Prompt user for 75-character text:
      print *, 'Enter NAME OF RUN (75-character string): '
      read *, runname

* Read description of model grid from program SETGRID output file:
      open (unit=11,file='setgrid.out',status='old')
      read (11, *)  xmin, ymin, dx, dy, ncols, nrows, izone
      npts = ncols * nrows

* Prompt for year, date, and hour of beginning of model run:
      print *, 'Enter START Year, Month, Day, Hour: '
      read *, startyr, month, day, hour
      startyr = mod (startyr, 1900)

* Make start year/date/hour time stamp:
      tstamp =  date2str (startyr, month, day, hour)

* Compute start hour relative to 00 hrs 1 January of start year,
      starthr = date2hrs (startyr, month, day, hour)

* Prompt for length of model run:
      print *, 'Enter LENGTH of model run (hrs.): '
      read *, runhrs

* Prompt for INPUT file paths, names, ID's, dt's:
      print *, ' INPUT files: '
      print *

      print *, ' BATHYMETRY map: '
      print *, '   Enter pathname: '
      read *, path
      print *, path
      print *, '   Enter file name: '
      read *, depname
      len1 = index(path,'/ ')
      len2 = index(path,' ')
      print *, len1, len2
      if (index(path,'/ ') .eq. 0) path = path(1:index(path,' ')-1) // '/'
      deppath = path

      print *, ' GRAIN SIZE map: '
      print *, '   Enter pathname: '
      read *, path
      print *, '   Enter file name: '
      read *, sizname
      if (index(path,'/ ') .eq. 0) path = path(1:index(path,' ')-1) // '/'
      sizpath = path

      print *, ' BACKGROUND CURRENT map: '
      print *, '   Enter pathname: '
      read *, path
      print *, '   Enter file name: '
      read *, bacname
      if (index(path,'/ ') .eq. 0) path = path(1:index(path,' ')-1) // '/'
      bacpath = path

      print *, ' TIDAL CURRENT maps: '
      print *, '   Enter pathname: '
      read *, path
      print *, '   Enter 2-character ID: '
      read *, idtide
      if (index(path,'/ ') .eq. 0) path = path(1:index(path,' ')-1) // '/'
      tidepath = path

      print *, ' WIND-DRIVEN CURRENT maps: '
      print *, '   Enter pathname: '
      read *, path
      print *, '   Enter 2-character ID: '
      read *, idwind
      print *, '   Enter dt (hrs.): '
      read *, dtwind
      if (index(path,'/ ') .eq. 0) path = path(1:index(path,' ')-1) // '/'
      windpath = path

      print *, ' WAVE FIELD maps: '
      print *, '   Enter pathname: '
      read *, path
      print *, '   Enter 2-character ID: '
      read *, idwave
      print *, '   Enter dt (hrs.): '
      read *, dtwave
      if (index(path,'/ ') .eq. 0) path = path(1:index(path,' ')-1) // '/'
      wavepath = path

* Prompt for INPUT file dt's:
      print *, ' OUTPUT files: '
      print *
      print *, '   Enter dt (hrs.) for SEDIMENT TRANSPORT maps: '
      read *, dtT
      print *, '   Enter dt (hrs.) for ACCUMULATED SEDIMENT maps: '
      read *, dtS
      print *, '   Enter dt (hrs.) for WAVE BREAKING maps: '
      read *, dtbrk

* Display model run parameters and write model parameters to 'sed94.f'
* input file 'sed94.in':
      open (unit=21,file='sed94.in')
      print 202, runname
      print *
      write (21, 202) runname
      write (21, *)
      print 204, idrun
      print *
      write (21, 204) idrun
      write (21, *)
      print 208, xmin, ymin, dx, dy, ncols, nrows, izone
      print *
      write (21, 208) xmin, ymin, dx, dy, ncols, nrows, izone
      write (21, *)
      print 210, startyr, starthr, tstamp, runhrs
      print *
      write (21, 210) startyr, starthr, tstamp, runhrs
      write (21, *)
      print 212, dttide, dtwind, dtwave
      print *
      write (21, 212) dttide, dtwind, dtwave
      write (21, *)
      print 214, dtT, dtS, dtbrk
      print *
      write (21, 214) dtT, dtS, dtbrk
      write (21, *)
      print 220, deppath, depname
      print *
      write (21, 220) deppath, depname
      write (21, *)
      print 222, sizpath, sizname
      print *
      write (21, 222) sizpath, sizname
      write (21, *)
      print 223, bacpath, bacname
      print *
      write (21, 223) bacpath, bacname
      write (21, *)
      print 224, tidepath, idtide
      print *
      write (21, 224) tidepath, idtide
      write (21, *)
      print 226, windpath, idwind
      print *
      write (21, 226) windpath, idwind
      write (21, *)
      print 228, wavepath, idwave
      print *
      write (21, 228) wavepath, idwave
      write (21, *)
      print 229

* Write names of time-dependent input files to 'sed94.in' for information only:
      write (21, 232)
      call names (tidepath,idtide,startyr,starthr,runhrs,dttide,'tid')
      write (21, *)
      write (21, 234)
      call names (windpath,idwind,startyr,starthr,runhrs,dtwind,'win')
      write (21, *)
      write (21, 236)
      call names (wavepath,idwave,startyr,starthr,runhrs,dtwave,'wave')

      close (21)

* Write parameter statement to be included in 'sed94.f':
      open (unit=22,file='sed94.inc')
      write (22, 240) ncols, nrows

      close (22)

      stop

202   format ('Name:' /  "'", a75, "'")
204   format ('Run ID:' /  "'", a2, "'")
208   format ('  xmin(m.)  ymin(m.)     dx(m)     dy(m)    columns'
      1        '      rows      izone' / 7i10)
210   format ('Startyr  Starthours   Start date/time          Runhours'/
      1         i2, 7x, i4, 9x, "'", a20, "'", 3x, i3)
212   format ('Input DT (hrs.):  Tidal current   Wind-driven current',
      1        '   Wave field'/
      1        t20, i2, t36, i2, t56, i2)
214   format ('Output DT (hrs.): Sed. transport  Acc. sed.  Wave breaking'/
      1        t20, i2, t36, i2, t42, i2)

220   format ('Bathymetry map (pathname, file name):' /
      1        "'", a40, "'  '", a20, "'")

222   format ('Grain size map (pathname, file name):' /
      1        "'", a40, "'  '", a20, "'")
223   format ('Background current map (pathname, file name):'/
      1        "'", a40, "'  '", a20, "'")
224   format ('Tidal current maps (pathname, file ID):' /
      1        "'", a40, "'  ", a2)
226   format ('Wind-driven current maps (pathname, file ID):' /
      1        "'", a40, "'  ", a2)
228   format ('Wave field maps (pathname, file ID):' /
      1        "'", a40, "'  ", a2)
229   format (/  " File 'sed94.in' contains a list of all input files " /
      1        "       required for the proposed model run." /)


232   format ('Tidal current input files: ')
234   format ('Wind-driven current input files: ')
236   format ('Wave field input files: ')
C     parameter (ncols=253, nrows=190)

240   format (t8, 'parameter (ncols=', i6, ', nrows=', i6, ')')

      end

      integer FUNCTION date2hrs (year, month, day, hour)
*     converts date and time to hours from 00:00 hrs 1 Jan. of input year.

      integer year, day, hour, days(11), hours
      data days /31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334/

      leap = 0
      if (mod (year, 4) .eq. 0) leap = 24

      if (month .eq. 1) hours = (day - 1) * 24 + hour
      if (month .eq. 2) hours = 744 + (day - 1) * 24 + hour
      if (month .ge. 3) hours = days(month-1) * 24 + leap + (day-1)*24 + hour
      date2hrs = hours

      return
      end

      character*20 FUNCTION date2str (year, month, day, hour)

*     creates date/time stamp character tstamp.

      character*3 mstring(12)
      character*2 n2(3), ystring, dstring, hstring
      integer year, day, hour
      data mstring / 'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
      1              'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec' /

      write (n2, '(i2.2)') year, day, hour
      read  (n2, '(a2)') ystring, dstring, hstring

      date2str = '19'//ystring//' '//mstring(month)//' '//dstring
      1       //' '//hstring//':00 UT'

      return
      end

      SUBROUTINE names (path, id, startyr, starthr, runhrs, dt, ext)
*     creates list of file names.

      logical done
      character*2 id, n2, yy
      character*3 ext
      character*4 n4, hhhh
```

```
      character*40 path
      character*60 name
      integer startyr, yr, starthr, dt, runhrs

      yr = mod (startyr, 1900)
      leap = 0
      if (mod (yr,4) .eq. 0) leap = 24

      endhr = starthr + runhrs
      done = .false.
      do 10 i = starthr, endhr, dt
              ihr = i
              if (i .gt. 8759+leap) ihr = i - (8760+leap)
              if (i .ne. ihr .and. .not. done) then
                      yr = yr + 1
                      done = .true.
              endif

             write (n2, '(i2)') yr
              read  (n2, '(a2)') yy
             write (n4, '(i4.4)') ihr
              read  (n4, '(a4)') hhhh

              len = index(path, ' ') -1
              name = path(1:len) // id // yy // hhhh // '.' // ext

              if (i .eq. starthr) then
                      write (21, '(3x, a60)') name
                      write (21, *) '           through'
              end if

              if (i .eq. 8759 + leap .and. i .ne. endhr)
     1                write (21, '(3x, a60)') name

              if (i .eq. 8760 + leap .and. i .ne. endhr) then
                      write (21, '(3x, a60)') name
                      write (21, *) '           through'
              end if

10            if (i .eq. endhr) write (21, '(3x, a60)') name

      return
      end
```

```
      PROGRAM sed94   ! 2-D sediment transport model with SEDTRANS92

     IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      integer dx, dy
      integer startyr, starthr, runhrs, endhr, elapsed
      integer dttide, dtwind, dtwave, dtT, dtS, dtbrk, dt
      integer year, hour, day, yrhour
      character*2 idrun, idtide, idwind, idwave
      character*7 hrs2str, label
      character*8 now, name
      character*9 today
      character*20 date2str, tstamp, logfile
      character*40 deppath, sizpath, bacpath, tidepath, windpath, wavepath
      character*20 depname, sizname, bacname, fname
      character*30 title1, title2
      character*75 runname

      include 'sed94.inc'
C     EXAMPLE of 'sed94.inc': "parameter (ncols=253, nrows=190)"
      dimension depth(ncols,nrows), gd(ncols,nrows)
      dimension bu(ncols,nrows), bv(ncols,nrows)
      dimension tu(ncols,nrows), tv(ncols,nrows)
      dimension wu(ncols,nrows), wv(ncols,nrows)
      dimension uz(ncols,nrows), cdir(ncols,nrows)
      dimension ht(ncols,nrows), per(ncols,nrows), dir(ncols,nrows)
      dimension Tx(ncols,nrows), Ty(ncols,nrows), S(ncols,nrows)
      dimension T(ncols,nrows)
      integer brk(ncols,nrows), brkhrs(ncols,nrows)
      integer xcode(ncols,nrows), ycode(ncols,nrows)
      integer xmin, ymin

      parameter (npts = ncols*nrows)
      data S / npts* 0.0 /
      data Tx / npts* 0.0 /, Ty / npts* 0.0 /
      data brk / npts* 0 / brkhrs / npts* 0 /

      pi = 4.0 * atan (1.0)
      deg2rad = 180.0 / pi

* Read model setup information from file 'sed94.in':
      call getsetup (runname, idrun, xmin, ymin, dx, dy, imax, jmax, izone,
     1               startyr, starthr, runhrs,
     1               dttide, dtwind, dtwave, dtT, dtS, dtbrk,
     1               depname, sizname, bacname, idtide, idwind, idwave,
     1               deppath, sizpath, bacpath, tidepath, windpath, wavepath)

* Read model parameters from file 'sed94.par':
      print *, " Read model parameters from file 'sed94.par'."
      call getpar (rkb, fract, rhos, rhow, z, iopt1, iopt2,
     1            conc0, towcd, towce, ws, prs, rkero)

* Read bathymetry map:
      print *, ' Read bathymetry from file ', depname
      call getmap (deppath, depname, imax, jmax, 1, depth, depth, depth)
      call setcodes (depth, imax, jmax, xcode, ycode)

* Read grain size map:
      print *, ' Read grain size distribution from file ', sizname
      call getmap (sizpath, sizname, imax, jmax, 1, gd, gd, gd)

* Read background current map:
      print *, ' Read background current from file ', bacname
      call getmap (bacpath, bacname, imax, jmax, 2, bu, bv, bv)

* Timing variables:
      dt = dttide                       ! define delta t (hours)

      endhr = starthr + runhrs          ! end hour of start year
      elapsed = -dt                     ! initialize

      open (unit=23, file='names')
* Perform time steps:
      do 100 ihr = starthr, endhr, dt          ! ihr = hour of start year
             elapsed = elapsed + dt            ! elapsed since start hour
             iprt = 0

*     Compute date and hour in current year:
             year = startyr
             yrhour = ihr
             call hrs2date (yrhour, year, month, day, hour)
             tstamp = date2str (year, month, day, hour)
             label = hrs2str (year, yrhour)
             name = idrun // label(1:6)

*     Write time stamp in log file:
             write (21, 202) ihr, elapsed, tstamp
             print 202, ihr, elapsed, tstamp

*     Get tidal current map:
             fname = idtide // label // 'tid'
             call getmap (tidepath, fname, imax, jmax, 2, tu, tv, tv)

*     Get wind-driven current map:
             if (mod(elapsed, dtwind) .eq. 0) then
                     fname = idwind // label // 'win'
                     call getmap (windpath, fname, imax, jmax, 2, wu, wv, wv)
             end if

*     Combine background current, tidal current and wind-driven current:
             call combine (bu, bv, tu, tv, wu, wv, uz, cdir, imax, jmax)

*     Get wave field map:
             if (mod(elapsed, dtwave) .eq. 0) then
                     fname = idwave // label // 'waves'
                     call getmap (wavepath, fname, imax, jmax, 3, ht,per,dir)
             end if
*     Apply SEDTRANS92 in each non-land cell:
             do 10 i = 1, imax                ! loop over columns
                     do 10 j = 50, jmax       ! loop over rows
                             Tx(i,j) = 0.0
                             Ty(i,j) = 0.0
                             brk(i,j) = 0.0

                             if (xcode(i,j).ne.6 .and. xcode(i,j).ne.7) then
      print *, gd(i,j)
                             if (gd(i,j) .le. 63.0) go to 10
                             grain = gd(i,j) * 1.0e-06
      print *, ' about to call sed92'
                             call sed92 (depth(i,j), grain,
     1                               uz(i,j), z, cdir(i,j),
     1                               ht(i,j), per(i,j), dir(i,j),
     1                               rkb, fract, rhos, rhow, iopt1, iopt2,
     1                               conc0, towcd, towce, ws, prs, rkero,
     1                               Tx(i,j), Ty(i,j), brk(i,j))
      print *, grain, Tx(i,j), Ty(i,j)
                             brkhrs(i,j) = brkhrs(i,j) + brk(i,j)
                             end if
10           continue

*     Account for sediment movement:
             do 20 i = 1, imax
                     do 20 j = 1, jmax
                     if (xcode(i,j) .eq. 6) go to 20

C                    Compute sediment transport magnitude T:
                             T(i,j) = sqrt (Tx(i,j)**2 + Ty(i,j)**2)

C                    Compute sediment transport divergence dTxdx, dTydy:
                             call divT (i, j, imax, jmax, dx, dy, Tx, Ty,
     1                                  dTxdx, dTydy, xcode, ycode)

*                    Compute sediment accumulation/depletion S:
                             deltat = dt * 3600.0d+00          ! dt in seconds
                             if (ihr .eq. starthr .or. ihr .eq. endhr)
     1                               deltat = dt * 1800.0d+00
                             S(i,j) = S(i,j) - (dTxdx + dTydy) * deltat

20                   continue

C Write output maps:
C    Sediment transport map:
                     title1 = runname
                     if (mod(elapsed, dtT) .eq. 0) then
                             iprt = 1
                             title2 = 'Sediment transport'
                             fname = idrun//label//'tpt'
                             call putmap (fname, title1, title2,
     1                               xmin, ymin, dx, dy, imax, jmax, izone,
     1                               year, yrhour, tstamp, 3, T, Tx, Ty)
                     end if

C    Sediment accretion/depletion:
                     if (mod(elapsed, dtS) .eq. 0) then
                             iprt = 1
                             title2 = 'Accumulated sediment'
                             fname = idrun//label//'acc'
                             call putmap (fname, title1, title2,
     1                               xmin, ymin, dx, dy, imax, jmax, izone,
     1                               year, yrhour, tstamp, 1, S, S, S)
                     end if

C    Wave breaking:
                     if (mod(elapsed, dtbrk) .eq. 0) then
                             iprt = 1
                             title2 = 'Wave breaking'
                             fname = idrun//label//'brk'
                             call putimap (fname, title1, title2,
     1                               xmin, ymin, dx, dy, imax, jmax, izone,
     1                               year, yrhour, tstamp, 2, brk, brkhrs, brkhrs)
                     end if

                     if (iprt .ne. 0) write (23, '(a8)') name


100          continue

* Finished. Print name of log file:
             logfile = idrun//'sed94.log'
             print *, ' Done. See log file ', logfile

* Make final entry in log file:
             call time (now)
             call date (today)
             write (21, 203) now, today
             close (21)       ! log file
             close (23)       ! names file

             stop

202          format (/ 1x, 'Hour = ', i4, '. Elapsed hours = ', i4, '. ', a20)
203          format (/ 1x, 'Done at ', a8, 5x, a9, '.  Whew!!')
             end
```