

SAS/GRAPH Techniques for Mapping Points, Lines and Small Areas

Mitchell S. Nussbaum, Wisconsin Department of Natural Resources

As Wisconsin's agency for environmental protection and natural resource management, the Department of Natural Resources (DNR) keeps track of a large number of entities and events, such as wells, forest fires and eagle's nests, that relate closely to specific geographic locations.

Maps are a natural way to display this data, but maps are not always easy to produce. Hand-mapping of large volumes of data is time-consuming and expensive. Geographic Information Systems (GIS) can map data quickly and fairly easily, but GIS software is expensive, requires specialized training, and is not readily available to most programs in DNR.

For many of DNR's needs, SAS/GRAPH[®] turns out to be the best available map-making tool. Almost everybody in the Department has access to SAS[®], and a large volume of DNR data is already managed or reported in SAS. SAS/GRAPH is less commonly used in DNR, but a SAS user will not need much additional training to use it.

SAS/GRAPH, however, is not perfect. We have found it necessary to develop a few tricks to meet the specific needs discussed in the paper, and these tricks have made map-making a bit less straightforward than we would like. Even with these adaptations, SAS as a mapping tool has some annoying quirks and limitations, which we will discuss later in the paper.

1. Mapping data by townships.

SAS/GRAPH comes with a map dataset showing Wisconsin county lines, which can be used as a base for choroplethic (shaded) maps of data summarized by county. These maps can be used to display some of DNR's statewide data, but they often hide important spatial patterns, and they are useless for generating maps that cover a single county.

A township-based map displays data with higher resolution. To support such maps, we used a previously-digitized file of township-corner coordinates to build a library of SAS map datasets

of Wisconsin townships. State and county outlines, digitized on the same coordinate system, were added to the library so maps could include a background of familiar boundaries.

Mapping data on this township base requires three principal steps:

1. Summarize the data, producing a dataset with at most one observation per township.
2. Build a map dataset. This involves two operations:
 - Subset the township map dataset, selecting the townships represented in the step 1 summary dataset
 - Append this dataset to an appropriate dataset of county boundaries.
3. Use PROC GMAP to generate a map from the dataset produced in step 1 and the map produced in step 2. Use the ALL option to plot the "empty" county boundaries as background.

Fig. 1 is an example of a map produced by this process.

2. Mapping individual data points

The most accurate view of data is often obtained by plotting individual data points, instead of shading in polygons to describe groups of data points. PROC GMAP is not specifically designed to make maps like this, but it can make them, with the help of ANNOTATE datasets.

Each observation in an ANNOTATE dataset is a command that performs some graphic operation. The FUNCTION variable determines the operation, and the X and Y variables specify where the operation will be performed. Other variables control the interpretation of X and Y, or modify the action specified by FUNCTION.

To produce a point map:

1. Start by generating an ANNOTATE dataset with one observation for each point in our original dataset. The ANNOTATE dataset contains the following variables:

X and Y

specify point locations in terms of the map's coordinate system.

FUNCTION

determines the graphic operation to be performed. For a point map, the operation is almost always "SYMBOL".

TEXT

specifies the symbol to be displayed. "PLUS" and "STAR" are frequently used; other legal values for this variable are shown in chapter 16 of the *SAS/GRAPH Software Reference* manual for Version 6.

COLOR

specifies the color used to draw the symbol

XSYS and YSYS

specify the reference system for the X and Y coordinates. Give XSYS and YSYS a value of '2' to interpret X and Y in the coordinate system of the map dataset.

TEXT and COLOR can be held constant for the entire dataset, or made dependent on the data values.

2. Build a map dataset, combining or subsetting map library datasets to include the outlines desired as background for the map. (The maps and accompanying ANNOTATE files in this paper use a rectangular coordinate system that can be plotted directly; if coordinates are specified in terms of latitude and longitude instead, use PROC GPROJECT to project the map and the ANNOTATE dataset. See page 1165 of the *SAS/GRAPH Software Reference*.)
3. Create a dummy dataset. GMAP will not run unless the DATA= parameter points to a valid dataset. To plot all the map polygons as "empty," create this dataset with a single observation, whose ID value does not match any value in the map dataset.
4. Finally, run PROC GMAP, with the CHORO command. Set MAP= the map dataset created in step 2. Specify the ALL option to plot all the polygons in the MAP dataset. Set DATA= the dummy dataset created in step 3. Set ANNO= the ANNOTATE dataset created in step 1. If you need legend lines to explain the map symbols used, create them with FOOTNOTE statements.

Fig. 2 is an example of a point map generated by this process.

3. Mapping lines.

Lines, e.g. rivers or roads, can also be mapped by an ANNOTATE dataset on a "choroplethic" background. Line mapping requires some additional values for the FUNCTION variable:

MOVE

moves the pen to the coordinates specified by X and Y.

DRAW

draws a line from the previous pen position to the coordinates specified by X and Y.

When FUNCTION = DRAW, the LINE variable can be used to specify the line type to be drawn. If LINE=1 (the default), the line will be solid. Values of 2 through 46 specify a variety of dash patterns. See chapter 16 of the *SAS/GRAPH Software Reference* for details.

Fig. 3 is a map of Wisconsin streams generated by this process. A MOVE command moves the pen to the mouth of a stream, and a series of DRAW commands draws a line upstream. The color of the line varies with the order of the stream (first-order streams cross the state line, second-order streams feed into first-order streams, etc.)

In Fig. 4, points and lines are displayed on a single map.

4. Wish List.

SAS/GRAPH is a useful tool for mapping points and lines, but it could be better. I would like to see:

1. *Explicit support of POINT and LINE maps by PROC GMAP.* Obviously, these maps would be easier to make if we could take full advantage of GMAP's functionality, instead of partially subverting it.
2. *Optional direct control of map scales.* Some users would like to map data on transparencies, to be overlaid them on paper maps from USGS or other agencies. Overlaying is impossible unless the SAS-produced transparency has the same scale as the paper map. SAS/GRAPH tries to fill its output page; I cannot find a simple way to override this tendency and specify a map scale.

3. A *point-in-polygon PROC*. A point-in-polygon procedure determines whether points in a dataset are inside or outside a specified polygon. Such a procedure is useful as a "cookie-cutter," for selecting the portion of a map or dataset that lies within a given area. The underlying algorithm for is well known; it simply needs to be packaged as a SAS PROC.

Trademark Notice

- * SAS/GRAPH and SAS are registered trademarks of SAS Institute, Inc., Cary, NC, USA.

Author

Mitchell S. Nussbaum
 Wisconsin Department of Natural Resources
 Box 7921
 Madison, WI 53707-7921
 Telephone: (608) 266-8831

Appendix A: Source code listings

Figure 1.

```
PROC FORMAT;
  VALUE FRQFMT 0='0'
    1-2 = '1 TO 2'
    3-5 = '3 TO 5'
    6-10 = '6 TO 10'
    11-50 = '11 TO 50'
    51-100 = '51 TO 100'
    101-HIGH = '101 AND UP';
DATA PESTDET;
  SET GWSAS.PESTDET;
  TWRNRG = TRSTWN || TRSRANGE;
* Build response dataset;
PROC FREQ DATA=PESTDET;
  TABLES TWRNRG / OUT=DFREQ;
PROC SORT DATA=PESTDET; BY TWRNRG;
PROC SORT DATA=DNRMAPS.TOWNS OUT=TOWNMAP;
  BY TWRNRG PTNUM;
*Select township polygons represented in response dataset;
DATA TOWNMAP;
  MERGE TOWNMAP (IN=A) PESTDET (IN=B);
  BY TWRNRG;
  IF A AND B;
DATA COUNTIES;
  SET DNRMAPS.COUNTIES;
  RENAME COUNTY=TWRNRG;
* Concatenate town & county maps;
DATA TOWNMAP;
  SET TOWNMAP COUNTIES;
LEGEND1 SHAPE=BAR(2,8);
PROC GMAP DATA=DFREQ MAP=TOWNMAP ALL;
  CHORO COUNT / DISCRETE LEGEND=LEGEND1;
  FORMAT COUNT FRQFMT.;
  ID TWRNRG;
```

```
LABEL COUNT='DETECTS';
TITLE1 C=RED 'PESTICIDE DETECTS';
TITLE2 C=RED 'AGGREGATED BY TOWNSHIP/RANGE';
PATTERN1 C=BLACK V=E;
PATTERN2 C=BLACK V=M1N0;
PATTERN3 C=BLACK V=M3N0;
PATTERN4 C=BLACK V=M2N60;
PATTERN5 C=BLACK V=M5X;
PATTERN6 C=BLACK V=S;
```

Figure 2.

```
*Select one observation per location for plotting;
PROC SORT DATA=GWSAS.PESTDET OUT=PESTDET; BY
  TRSTWN TRSRANGE TRSSECT;
DATA PESTDET;
  SET PESTDET;
  BY TRSTWN TRSRANGE TRSSECT;
  IF FIRST.TRSTWN OR FIRST.TRSRANGE OR
  FIRST.TRSSECT;
*Build ANNOTATE dataset, one SYMBOL command for each
  point;
DATA ANNOT;
  SET PESTDET;
  XSYS = '2'; YSYS = '2';
  X = EASTTM; Y = NORTM;
  FUNCTION = 'SYMBOL';
  TEXT = 'PLUS';
  COLOR = 'BLACK';
*Create dataset with a dummy observation for PROC GMAP;
DATA DUMMY;
  COUNTY = '99';
  COUNT = 0;
PROC GMAP DATA=DUMMY MAP=DNRMAPS.COUNTIES
  ALL ANNOTATE=ANNOT;
  CHORO COUNT / NOLEGEND;
  ID COUNTY;
  TITLE1 C=RED 'PESTICIDE DETECTS';
```

Figure 3.

```
*Build ANNOT1 dataset to plot first-order streams;
DATA FIRSTORD;
  SET WATER.STRPTS;
  IF SUBSTR(INDEX3,1,5) = '00000';
  PROC SORT DATA=FIRSTORD OUT = FIRSTORD;
  BY MINOR INDEX1 INDEX2;
DATA ANNOT1;
  SET FIRSTORD;
  LENGTH FUNCTION $ 8;
  BY MINOR INDEX1 INDEX2;
  IF SUBSTR(INDEX1,1,5) ^= '00000';
  XSYS = '2'; YSYS = '2'; WHEN = 'A';
  Y = NORTH; X = EAST;
  IF FIRST.INDEX1 OR FIRST.MINOR THEN DO;
  IF NOT LAST.INDEX1 THEN DO;
*Move pen to mouth of stream
  FUNCTION = 'MOVE';
  OUTPUT;
  END;
  END;
  ELSE DO;
```

```

*Going upstream draw line connecting stream points;
FUNCTION = 'DRAW';
COLOR = 'BLUE'; LINE=1; *line is blue & solid;
OUTPUT;
END;
*Build ANNOT2 dataset to plot second-order streams;
DATA SECORD;
SET WATER.STRPTS;
IF SUBSTR(INDEX2,1,5) ^= '00000' AND
SUBSTR(INDEX4,1,5) = '00000';
PROC SORT DATA=SECORD OUT = SECORD;
BY MINOR INDEX1 INDEX2 INDEX3;
DATA ANNOT2;
SET SECORD;
LENGTH FUNCTION $ 8;
BY MINOR INDEX1 INDEX2 INDEX3;
XSYS = '2'; YSYS = '2'; WHEN = 'A';
Y = NORTH; X = EAST;
IF FIRST.INDEX1 OR FIRST.INDEX2 OR FIRST.MINOR
THEN DO;
    FUNCTION = 'MOVE';
    OUTPUT;
END;
ELSE DO;
    FUNCTION = 'DRAW';
    COLOR = 'LIME'; LINE=3; *line is lime-colored & dashed;
    OUTPUT;
END;
*Build ANNOT3 dataset to plot third-order streams;
*(follows pattern for first and second order; SAS code omitted
to conserve space);
DATA THRDORD;...DATA ANNOT3;...
*Concatenate ANNOT1,2,3 to form single ANNOTATE
dataset;
DATA ANNOT;
SET ANNOT1 ANNOT2 ANNOT3;
DATA DUMMY;
PID = 99;
COUNT = 1;
PROC GMAP DATA=DUMMY MAP=DNRMAPS.RSBASINS
ALL;
ID PID;
CHORO COUNT/DISCRETE CEMPTY=BLACK
NOLEGEND ANNOTATE=ANNOT;
TITLE1 C=RED 'Wisconsin Streams';
TITLE2 C=BLUE 'first order, ' C=LIME
'second-order ' C=BLACK 'and ' C=GOLD
'third order';
FOOTNOTE1 C=RED F=ITALIC
"stream traces" derived from Master Waterbody File"
PATTERN1 C=BLACK V=E;
RUN;

```

Figure 4

```

*Select streams in Wolf River basin;
*STREAMS dataset describes streams as a whole
DATA STRSEL;
SET WATER.STREAMS; IF MINOR = '082';
PROC SORT DATA=STRSEL; BY WBID;
PROC SORT DATA=WATER.TRACE OUT = TRACE;

```

```

BY WBID RIVMILE;
*TRACE contains coordinates of pts along streams;
* Select coordinates of streams in Wolf R. basin;
DATA TRACE;
MERGE STRSEL (IN=A) TRACE (IN=B); BY WBID;
IF A AND B;
*Build ANNOTATE dataset to plot streams, dams & rapids;
DATA ANNOT;
SET TRACE;
LENGTH FUNCTION $ 8;
LENGTH COLOR $ 6;
LENGTH TEXT $ 20;
BY WBID RIVMILE;
XSYS = '2'; YSYS = '2'; WHEN = 'A';
Y = NORTH; X = EAST;
IF FIRST.WBID THEN DO; FUNCTION = 'MOVE'; OUTPUT;
END;
ELSE DO;
*Wolf R. is third-order -- plot as red solid line;
IF ORDER = 3 THEN DO; COLOR = 'RED'; LINE=1; END;
*Plot other streams in basin as blue dashed lines;
ELSE DO; COLOR = 'BLUE'; LINE=3; END;
FUNCTION = 'DRAW';
OUTPUT;
END;
IF WBTYPE = '3' THEN DO;
FUNCTION = 'SYMBOL'; *plot green star at dams & rapids;
TEXT = 'STAR'; COLOR = 'GREEN';
OUTPUT;
END;
DATA DUMMY;
PID = 99; COUNT = 1;
*Build map dataset showing Wolf R basin boundaries;
DATA WOLFMAP;
SET DNRMAPS.RSBASIN; IF MINOR='082';
PROC GMAP DATA=DUMMY MAP=WOLFMAP ALL;
ID PID;
CHORO COUNT/DISCRETE CEMPTY=BLACK NOLEGEND
ANNOTATE=ANNOT;
TITLE1 C=BLACK 'The Wolf River Basin';
TITLE2 C=BLACK 'showing all streams in Master Waterbody
File';
TITLE3 C=RED 'Wolf River shown as red solid line';
TITLE4 C=RED 'Tributary streams shown as blue dashed
lines';
FOOTNOTE1 c=green '*:Dams and rapids';
PATTERN1 C=BLACK V=E;

```

Figure 1

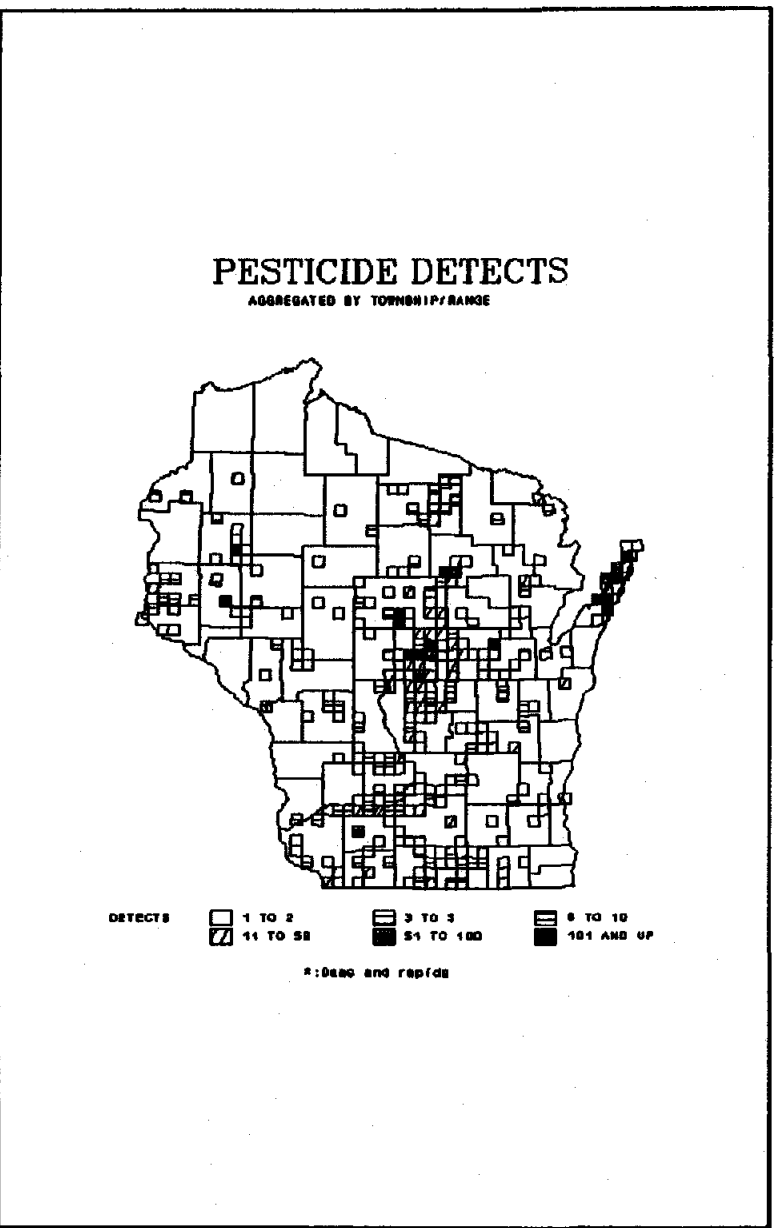


Figure 2

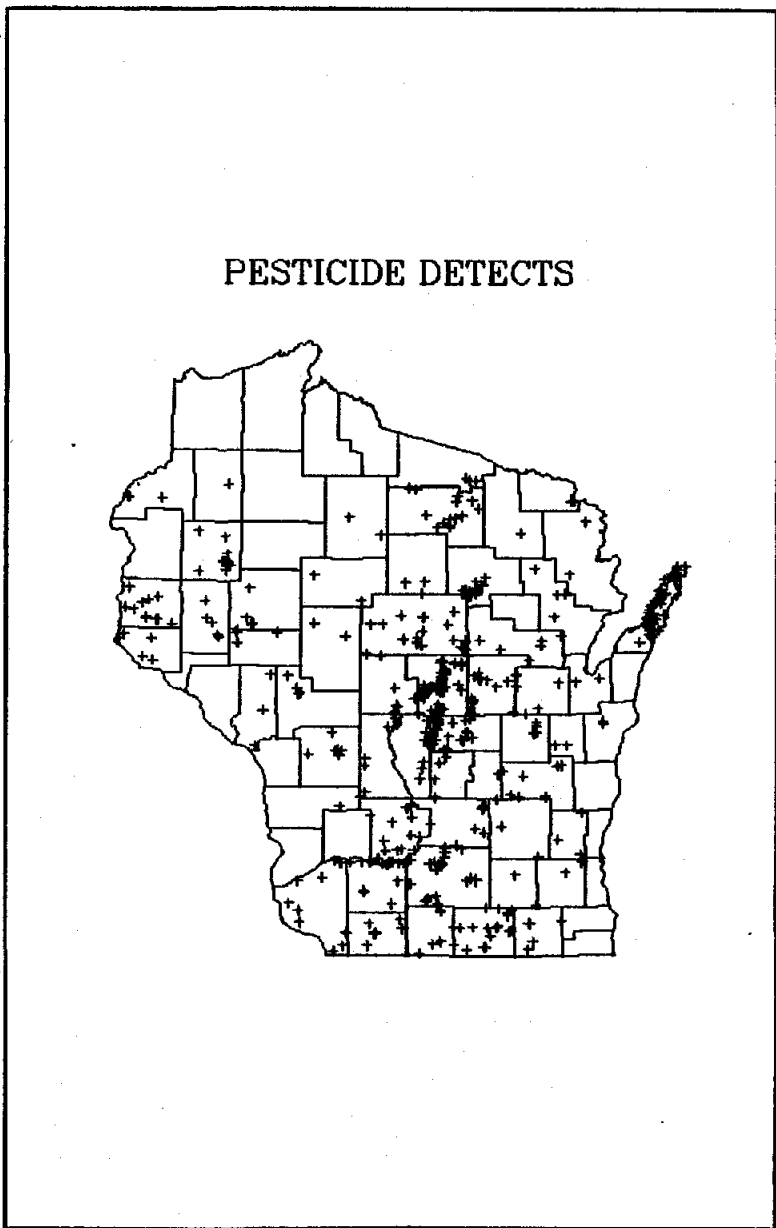


Figure 3

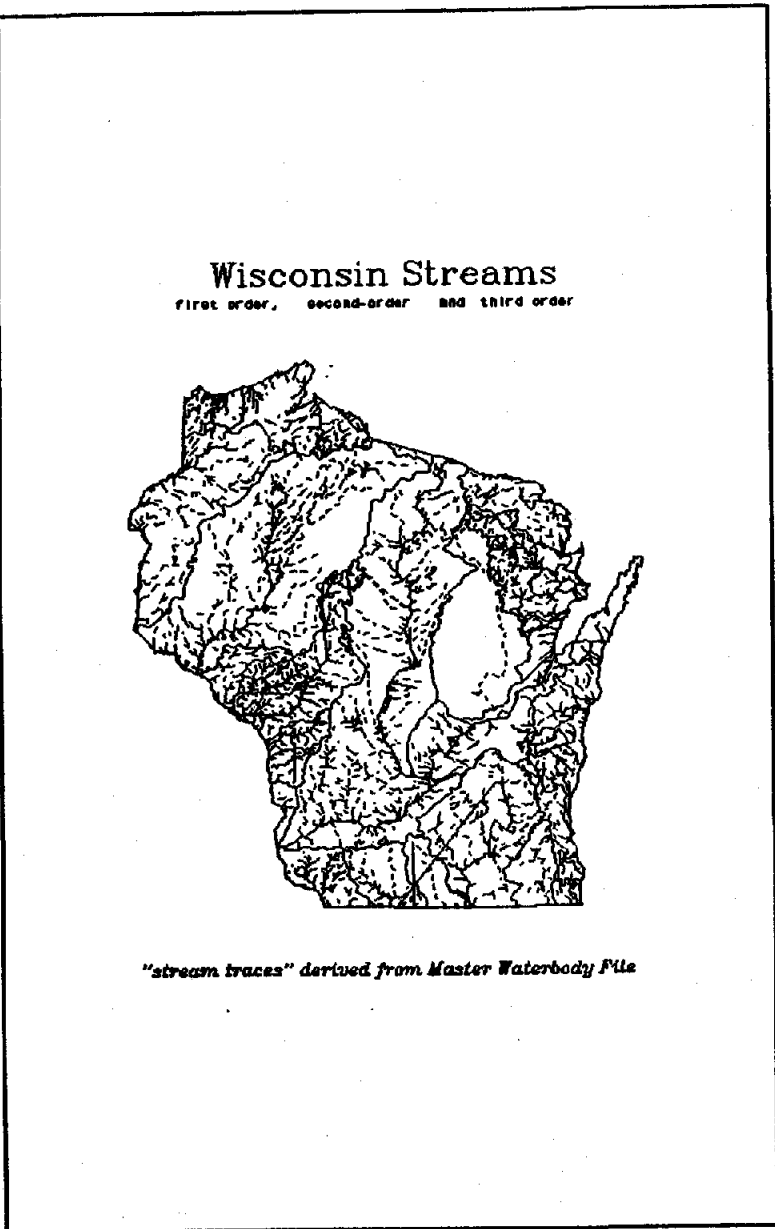


Figure 4

