GAMS / GAMSMAP: A Library of GAMS Routines to Create Customized MapInfo Maps *

Armin Pruessner

April 25, 2002

Contents

*GAMSMAP is available online at `http://www.gams.com/contrib/gamsmap/index.htm`

1 CHANGES

April 25, 2002

1. Added `--colorfile` option. This option allows users to specify a file to assign colors to a set of regions. For example, suppose users wish to color all US states with different colors according to colors specified in a file. This is achieved by inputting a file using the `--colorfile=(filename)` option. Users list the region name and the IRGB value. The IRGB value for valid colors is listed in the file `rgbcolor.txt` which is located in the `SampleData/` directory. This feature is described in the Advanced Features section of this manual.

2. Sample colors files for US states and world countries have been added to the `Sampledata/` directory. The files are called `states_col.txt` and `world_col.txt`.

April 16, 2002

1. Changed structure of GAMSMAP directory so that data files needed are extracted during runtime from a zip archive called GMSMAPDB.zip. Size of GAMSMAP directory is minimized and unnecessary files are not extracted. Files used are placed in the GAMS 225* scratch directory and deleted upon completetion of execution.

January 18, 2002

1. Added multiple scaling options. The user may specify `-- scaleopt=0` or `--scaleopt=1`, where 0 specifies that scaling for numerical data is based on a $[0, max]$ scale and 1 specifies a $[min, max]$ scale. The default is 0.

2. Added a tutorial file `Tutorial.pdf` to the documentation section under (`../GAMSMAP/Documentation/`).

3. Added a new region data sets for the `--data` option. Users can specify `world` as a region which will create a world map of all countries. The country identifier codes (3 letter) are available in the file `world_n.txt`.

4. Added new point data sets for the `--data` option. Users can specify either `uscaps` or `worldcaps` to specify US state capitals or capitals of countries respectively.

5. Added new sample numerical data files: `world_lit.txt` which contains 1994 data on literacy rates around the world, and `worldcaps_pop.txt` which contains the population of capital cities around the world.

January 8, 2002

1. Changed −− `size` and −− `color` option for point maps when datafiles are used where all numerical values are equal. In this case, the point will be shown using the −− `color` and −− `size` specified during input, not the numeric color −− `colp` or −− `coln` and the size.

2. Changed −− `color` option for regions maps when datafiles are used where all numerical values are equal. In this case, the region in the −− `datafile` will be shown using the −− `color` instead of the numeric color −− `colp` or −− `coln`. The other regions will be shown but not be colored.

3. Changed −− `color` option for link maps when datafiles are used where all numerical values are equal. In this case, the link in the −− `datafile` will be shown using the −− `color` instead of the numeric color −− `colp` or −− `coln`. The link width will be as specified in −− `width`.

4. Point offset option for point maps. If a point is to be plotted not at it's actual coordinates, but offset. The user can specify −− `xoffset=offset` in x direction and −− `yoffset=offset` in y direction. This is useful when plotting different types of data for single points.

5. Point offset option for links. This is identicial to the option for point maps.

6. Changed Mapinfo information about links. The information now reported is "Origin", "Destination" and numeric value. For bidirectional links, the width and coloring is in the direction of the link with the larger absolut value.

2 INTRODUCTION AND INSTALLATION

GAMSMAP is a collection of GAMS routines and data files to create customized MapInfo maps depicting regions, data points and networks. With GAMSMAP the user can create a wide variety of maps showing locations of specific regions or data coordinates or dependence of regions and coordinates on numerical values. Furthermore, flow from one set of data points to another set can be visualized.

The routines in GAMSMAP are highly modular. That is, customized maps can easily be created by building individual map components, for example states, counties, capitals, and then merging them. This feature allows maps to be changed with minimal effort, if for example only a minor component is to be changed from one map to another.

The reader is assumed to be familiar with the more common GAMS commands. If not, the user is referred to the GAMS User's Manual [1].

2.1 Installation

A GAMS system (Version 2.50, Distribution 21.0 or later) is required to run all routines and a Mapinfo system (Version 6.5 or later) to view all resulting maps. Earlier version of GAMS do not support the `--option` command line feature to initialize environment variables. As we will see later, this feature is essential to using the GAMSMAP routines.

To install, the user copies the files *GMSMAP.zip* to the desired location and unzips either using gmsunzip or any other utility. The directory contains subdirectories with necesary mapping data, as well as all the GAMS run files to create the customized maps.

2.2 Main Routines

The main GAMSMAP routines consist of three GAMS files, with which the user can create all types of custom maps.

- **REGIONS.GMS**: creates region maps depicting US states and world countries.
- **POINT.GMS**: creates coordinate maps depicting centroids of states, centroids of world countries, US state capitals, or world country capitals.
- **LINK.GMS**: creates link maps depicting flow from point to point, where a point is as defined previously. The object from which flow originates must be of the same type as the object where the flow terminates.

The concept behind GAMSMAP is that the user creates component maps (MapInfo .tab files) of regions, points, and links and then merges them together by opening several different table files in MapInfo simultaneously. This will be discussed in §2.

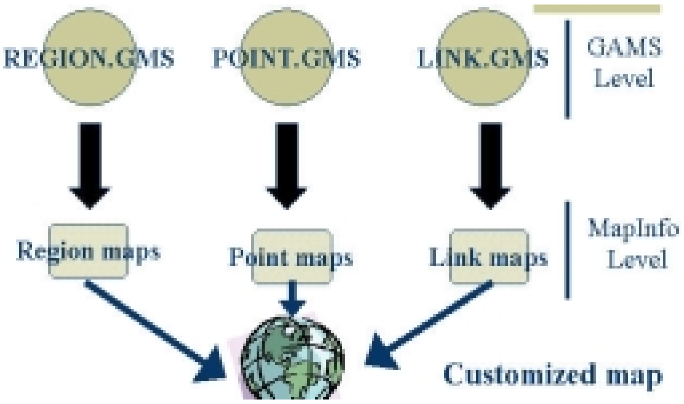

Figure 1: GAMSMAP Overview: Merging of Components

2.3 GMSMAPDB.zip: Data Libraries and Support Files

All necessary data files and supporting GAMS files are located in the database file GMSMAPDB.zip. Upon execution of one of the three main mapping routines (region.gms, point.gms, or link.gms), the necessary data and support files are extracted from GMSMAPDB.zip, placed in the GAMS 225* scratch directory, and then deleted upon completion of the GAMS run.

The runtime extraction of data and support files eliminates the need for large data sets to be stored on the user machine.

3 GENERAL USAGE

3.1 The GAMSMAP Command

All routines in GAMSMAP library are called using the following command at the command prompt (or an anologous command in the process window of the GAMS IDE):

> *gams (gams filename) −−option1=x −−option2=y −−option3=z ...*

The strings `−−option1=...` are user specified options, such as width of border, color, etc. These are generally specific to the GAMSMAP routine used, although many are similar for all three files. Some may be ommited, in which case, the program uses a default value. Each GAMS routine has several user input option to customize the map desired. See §5 on each specific GAMSMAP file for details.

3.2 Subcomponents and Layering of Component Maps

If a complex map consisting of several regions, points, and links is desired the user must call a GAMSMAP routine for each object seperately. That is, for each region (for example states, counties, etc.), point (plants, region centroids) and links a seperate component maps is created.

For example, suppose a map with the U.S. states, counties, and coordinate points of the state centroids is desired. The user would run the following routines to get the following component mapping files (.tab):

- REGION.GMS → states.tab
- REGION.GMS → counties.tab
- POINT.GMS → plants.tab

To assemble the final map, the user simply opens each table in MapInfo and layers them as desired. Since each object (states, counties, centroids) is a seperate component, it is very simple to make changes to the final map. The user can hide certain objects, without having to rerun a GAMSMAP routine. We will give examples on how to create more complicated maps by layering individual components in §6.

4 USE OF NUMERICAL DATA

All three main routines support the use of numerical data in the maps created. We will give a brief overview of the text data files format that is supported by GAMSMAP for using numerical data.

4.1 Region Shading, Point Sizes, and Link Widths

For regions, numerical data can be represented by varying degrees of shading. The color of the shading depends on the sign of the numerical value associated with the particular region. The user may specify a color associated with positive and negative values - if none is specified, default values of green for positive and red for negative values are assigned. The scale for the shading is automatically adjusted to the range of the data. That is, the maximum absolute value of both positive and negative values receives the deepest shade of the particular color (positive or negative), whereas the smallest absolute value the lightest shade. Shading for values in between are assigned on a linear scale, although the user can customize the shading scale. This will be discussed in the Advanced Features section.

For points, numerical data is represented by the size of the point or through the use of shading to color the point interior. For the size option a larger numerical value corresponds to a larger point size, and a smaller value to a smaller point size. Similar rules as for region maps are followed for positive and negative values in terms of coloring. Furthermore, point sizes are assigned similarly on a linear scale between the maximum and minimum absolute values. For the color option, the user specifies the point size and the interior is then shaded according to the numerical value specified (see the region routine description for details).

For links between data points, the user specifies starting and end points, as well as a numerical value associated with the link. The value is represented by the arrow width, again displaying larger values by wider arrows and vice versa. Also, positive and negtaive values are displayed using a user-specified coloring scheme or using default values for regions and points.

4.2 Textfile Option

If data is extracted from other sources, for example from Excel data, the data can be read from textfiles. The textfiles generally have the format

```
Object ID   Value.
```

For example,

```
      VA    97.2
(or) FRA   -54.1
```

For link data files, the format is similar, except that point identifiers for both origin and destination must be specified:

```
Origin ID   Destination ID   Value.
```

For example,

```
       TX    NM    97.2
(or)  JPN   AUS   -54.1
```

5 THE GAMSMAP ROUTINE --OPTIONS

In this section we will list all of the user options for each of the three main GAMSMAP routines. Some have default values.

5.1 Region.gms

`−−color`	If no datafile is included, then −−color specifies the region interior color. If a −−datafile is included, then −−color specifies the region interior color of those regions not specified in the datafile. (*default = lightyellow*)
`−−colp`	Applies only if a −−datafile is specified. Shading of positive numeric values associated with each region in the datafile. Can only be a base color. Base colors are red, blue, green, yellow, cyan, and magenta. (*default = green*)
`−−coln`	Applies only if a −−datafile is specified. Shading of negative numeric values associated with each region in the datafile. Can only be a base color. Base colors are red, blue, green, yellow, cyan, and magenta. (*default = red*)
`−−colorfile`	Filename of color file. If users wish to assign colors to specific regions without having numerical data, users can include a file having the region name and its IRGB value. The IRGBN value is a unique integer identifier combining the three RGB values. Valid colors and their respective IRGB values are listed in the file `SampleData/rgbcolor.txt`. A sample `−−colorfile` entry for the color blue is: `JPN 255`. (*default = none*)
`−−data`	Region type. Can be either states or world. (*default = states*)
`−−datafile`	Data filename. Text file (*default = none*)
`−−noplot`	Object ID(s) of regions to be ignored. Default is "AK,HI, Canadian states". Regions specified in −−noplot are ignored and not shown in map. If a region is specified in both −−plot and −−noplot, then −−noplot overrides. If a region specified in −−noplot does not exist in the total set of that type (for example TX is not a control area), then it is ignored. (*default = AK, HI, Canadian states*)

`−−plot`	Object ID(s) to be plotted. If the −−plot option is included only those regions specified in −−plot are included. If a region in −−plot is specified for which no data exists, the region is ignored, but a warning message appears in status.log file. If a region is specified in both −−plot and −−noplot, then −−noplot overrides. If a region specified in −−plot does not exist in the total set of that type (for example TX is not a control area), then it is ignored. (*default = all regions*)
`−−scale`	Filename. GAMS scaling file for custom scaling of region shading, point sizes, and link widths. (*default = none*)
`−−scaleopt`	Integer (1 or 0). Scaling mode. 0 determines scaling on the range [0,max] and 1 on the range [min,max]. (*default = 0*)
`−−shade`	Region interior type. The user may choose from solid, fade, lightfade, none, where solid is completely filled and the others are gradually more faded until transparent. (*default = solid if no −−color is specified and solid if a −−color is specified. If a data file is included, those regions listed in the data file have a solid shading.*)
`−−tabfn`	Output filename (*default = region.tab*)
`−−width`	Region border width. Values can be between 1 and 7. (*default = 1*)

5.2 Point.gms

`−−color`	If no datafile is included, then −−color specifies the interior color of the point. If a −−datafile is included, then −−color specifies the interior color of those points not specified in the datafile. (*default = lightyellow*)
`−−colp`	Applies only if a −−datafile is specified. Shading of positive numeric values associated with each point in the datafile. Can only be a base color. Base colors are red, blue, green, yellow, cyan, and magenta. (*default = green*)
`−−coln`	Applies only if a −−datafile is specified. Shading of negative numeric values associated with each point in the datafile. Can only be a base color. Base colors are red, blue, green, yellow, cyan, and magenta. (*default = red*)
`−−data`	Point type. Can be either states, countries, uscaps, or worldcaps. For states and countries the option refers to region the centroid. (*default = plants*)
`−−datafile`	Data filename. Text file. (*default = none*)
`−−noplot`	Object ID(s) of points to be ignored during plotting. Default is "AK,HI, Canadian states". Points specified in −−noplot are ignored and not shown in map. If a point is specified in both −−plot and −−noplot, then −−noplot overrides. If a point specified in −−noplot does not exist in the total set of that type (for example TX is not a control area), then it is ignored. (*default = AK, HI, Canadian states*)
`−−plot`	Object ID(s) of points to be plotted. If the −−plot option is included only those points specified in −−plot are included. If a point in −−plot is specified for which no data exists, the point is ignored, but a warning message appears in status.log file. If a point is specified in both −−plot and −−noplot, then −−noplot overrides. If a point specified in −−plot does not exist in the total set of that type (for example TX is not a control area), then it is ignored. (*default = all regions*)
`−−scale`	GAMS scaling file for custom scaling of region shading, point sizes, and link widths. (*default = scaling.gms*)
`−−scaleopt`	Integer (1 or 0). Scaling mode. 0 determines scaling on the range [0,max] and 1 on the range [min,max]. (*default = 0*)

`−−size`	Point size. Applies only if no −−datafile is included or if a −−datafile is included, the −−type=color. Size ranges from 0-48. (*default = 0*)
`−−tabfn`	Output filename. (*default = points.tab*)
`−−type`	Type of numerical value representation. Color or size. Applies only if a −−datafile is included. User may choose from −−type=size (numerical values are shown by size) or –type=color (numerical values are shown by shading). (*default = size*)
`−−xoffset`	Offset amount of x–coordinate from coordinate point. Useful if plotting several points at one location. (*default = 0*)
`−−yoffset`	Offset amount of y–coordinate from coordinate point. Useful if plotting several points at one location. (*default = 0*)

5.3 Link.gms

`−−color`	Applies only if all numerical values are set to 1 (i.e. no numerical data). The option −−color specifies the color of the links between points specified in the −−datafile. (*default* = *black*)
`−−colp`	Applies only if numeric values are specified in the −−datafile. Shading of positive numeric values associated with each arrow in the datafile. Can only be a base color. Base colors are red, blue, green, yellow, cyan, and magenta. (*default* = *green*)
`−−coln`	Applies only if numeric values are specified in the −−datafile. Shading of negative numeric values associated with each arrowt in the datafile. Can only be a base color. Base colors are red, blue, green, yellow, cyan, and magenta. (*default* = *red*)
`−−data`	Object type connected by arrows. Can be either states, world, uscaps, or worldcaps. For states and world, the parameter refers to region centroid. (*default* = *states*)
`−−datafile`	Data filename. A −−datafile must be included. The −−datafile contains the starting and ending point ID of the link and possibly a numerical value associated with that link. The −−datafile is a .txt file. Input example: $point\_ID1 \quad point\_ID2 \quad 99.97$ If no numerical values are desired, then set all values to 1. Can also be a Dbase file. In this case −−dbindex, –dbfield, dbfield2, must be indicated to specify origin, destination, and value. (*default* = *links.txt*)
`−−noplot`	Pair of object ID(s) of links to be ignored during plotting. Pairs specified in −−noplot are ignored and not shown in map. If a pair is specified in both −−plot and −−noplot, then −−noplot overrides. If a pair specified in −−noplot does not exist in the total set of that type (for example TX is not a control area), then it is ignored. Pairs are seperated by a .dot. For example TX.NM (*default* = *none*)

`−−plot`	Pair of object ID(s) of links to be plotted. If the −−plot option is included only those links specified in −−plot are included. If a pair in −−plot is specified for which no data exists, the pair is ignored, but a warning message appears in status.log file. If a pair is specified in both −−plot and −−noplot, then −−noplot overrides. If a pair specified in −−plot does not exist in the total set of that type (for example TX is not a control area), then it is ignored. Pairs are seperated by a .dot. For example JPN.RUS (*default = all regions*)
`−−scale`	GAMS scaling file for custom scaling of region shading, point sizes, and link widths. (*default = scaling.gms*)
`−−tabfn`	Output filename. (*default = links.tab*)
`−−width`	Width of the link. The −−width option applies only if no numerical values are given in the −−datafile. (*default = 1*)
`−−xoffset`	Offset amount of link x–coordinates from original origin and destination points. Useful if plotting several links at one location. (*default = 0*)
`−−yoffset`	Offset amount of link y–coordinates from original origin and destination point. Useful if plotting several links at one location. (*default = 0*)

6 EXAMPLES

We shall illustrate several examples and show how to create the component maps necessary for each one. The examples should also show how to make use of many of the GAMSMAP `−−options`.

6.1 Example 1 - Regions

The first example consists of a world map with specific countries highlighted in red. For this example, we will highlight Canada, Russia, and Australia. This will require two GAMSMAP calls to the region.gms routine. The first is:

```
> gams region −−data=world −−color=lightyellow −−tabfn=world
```

This createds the map *world.tab*, depicting the countries of the world with interior color lightyellow. We now want to create a second map highlighting Canada, Russia, and Australia.

```
> gams region −−data=world −−color=red −−plot=”CAN,RUS,AUS”
−−tabfn=countries
```

The second region map created is *countries.tab*, consisting only of those countries defined in `--plot`. The region interior color is red. The final map, constructed by opening both .tab files in MapInfo is shown in Figure 2.

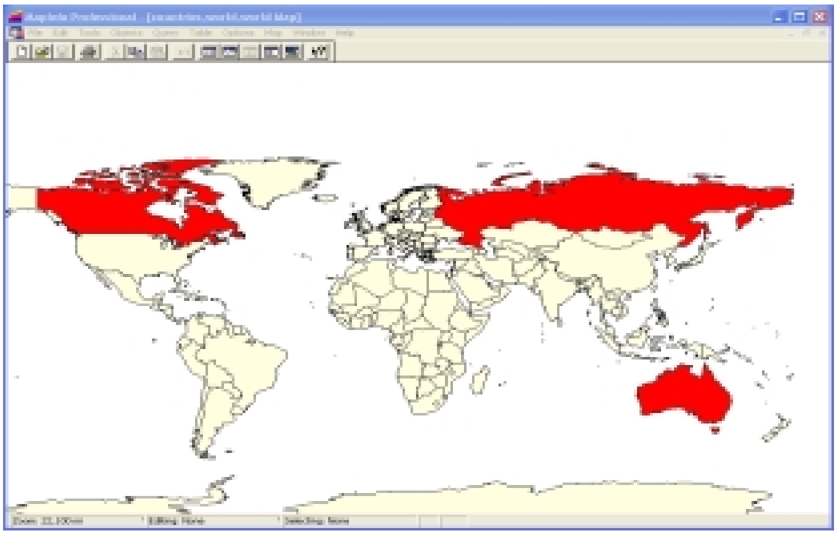

Figure 2: Example 1: Region Map (No Numerical Data)

6.2 Example 2 - Points and Regions

The second example illustrates the use of the point.gms routine. It consists of the region map of states and the point map of the state capitals. The required GAMSMAP calls are:

```
> gams region r=0 --data=states --color=lightyellow
--tabfn=states
```

This creates the state region map called *states.tab* with a region border width of size 1 (default) and interior color of lightyellow. We then create the point map file for the state capitals:

```
> gams point r=0 --data=uscaps --color=red --size=10
--tabfn=caps
```

This creates the US state capitals as size 10 points in red. The output filename is *states.tab*. As before, the subcomponent maps can be assembled in MapInfo. The resulting map is in Figure 3.

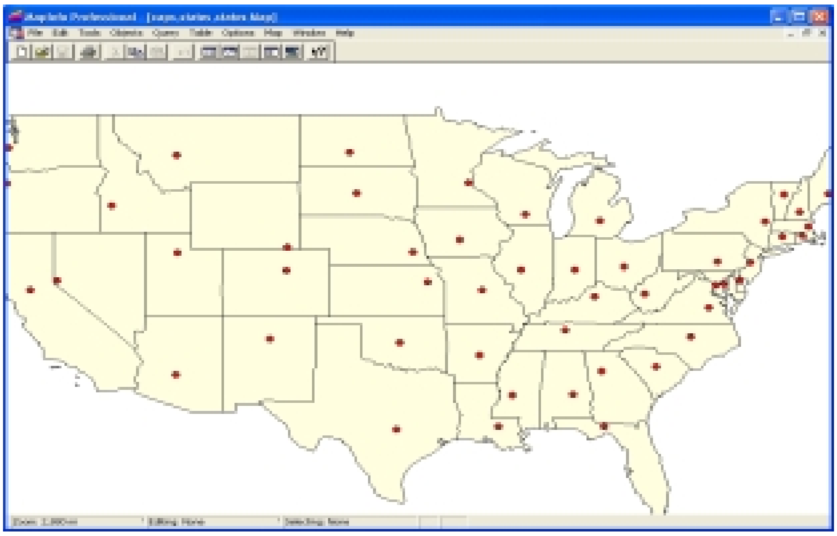

Figure 3: Example 2: Point Map (No Numerical Data)

6.3 Example 3 - Regions With Numerical Data

Our examples so far have been limited to non-numeric data maps. Let us now consider a control area region map with dependency on numerical data.

Suppose we want to create a map depicting world literacy rates around the world. We have a data file in the directory `SampleData/` containing numerical data (world_lit.txt), the country codes, and the numerical data (taken from a MapInfo data set from 1994). The format is:

```
     ''BHS''  90.0
(or) ''BHR''  77.0
```

The GAMSMAP command is:

> *> gams region −−data=world −−datafile=SampleData/world_lit.txt −−colp=red −−tabfn=worldlit*

The data file is called `world_lit.txt` and is located in the sample directory `SampleData/`. The full path `SampleData/world_lit.txt` must be specified. The actual mapping file will be called *worldlit.tab*. The color shading will be red, as defined by the `−−colp=red` statement. This refers to the shading of the positive numerical values. A similar command `−−coln` exists for shading negative numerical values.

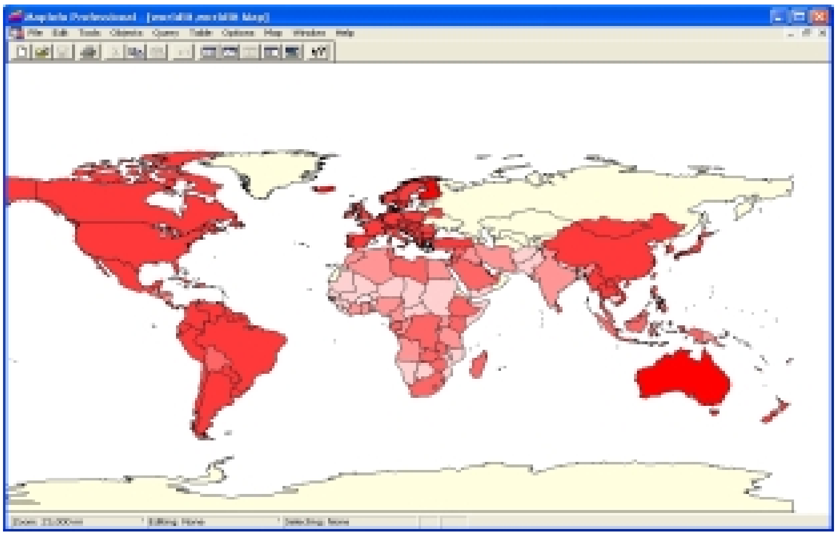

Figure 4: Example 3: Region Map (Numerical Data)

6.4 Example 4 - Points With Numerical Data

Let us now look at point maps with numerical data. We have two options in terms of displaying numerical data for points. This is specified by the `−−type` option, which takes on values of *color* or *size* (default). The size option varies the size of the point based on numeric data, whereas the color option varies the shading interior of the point. For the latter option it is useful to set the `−−size` option to values of 10 or higher in order to distinguish the different shading.

Suppose we want to create a map depicting countries in the world as well as the capitals of the countries, where the size of the points depicting the capitals show the respective populations of these cities. This will require two component maps.

Let us first create a region map depicting the world countries. The command is:

```
> gams region −−data=world −−tabfn=world
```

This creates a .tab file called world.tab. Now let us create a point map showing the points:

```
> gams point −−data=worldcaps −−colp=yellow
−−datafile=SampleData/worldcaps_pop.txt −−tabfn=worldcappop
−−scale=SampleData/scaling_new.gms
```

This command creates a point map using the GAMS file point.gms. We specify that our data set is world capitals using the `−−data=worldcaps` option.

Our data set comes from the data file `worldcaps_pop.txt`, located in the sample data file directory `SampleData/`. Our positive coloring of the points will be yellow and our map file (.tab) will be called *worldcappop.tab*. We have also defined a different scaling than the default scaling, by specifying −−`SampleData/scaling_new.gms`, which tells GAMS to look for a new scaling file called `scaling_new.gms` in the sample data directory `SampleData/`. The user may create custom scaling files as needed. In this case the new scaling file has modified values for the point size. Custom scalings are discussed in §7 on advanced GAMSMAP features.

For the color option the command is:

```
> gams point −−data=worldcaps −−colp=yellow −−size=10
−−datafile=SampleData/worldcaps_pop.txt −−type=color
−−scale=SampleData/scaling_new.gms −−tabfn=worldcappop
```

The resulting maps are show in Figures 5 (size) and 6 (color).

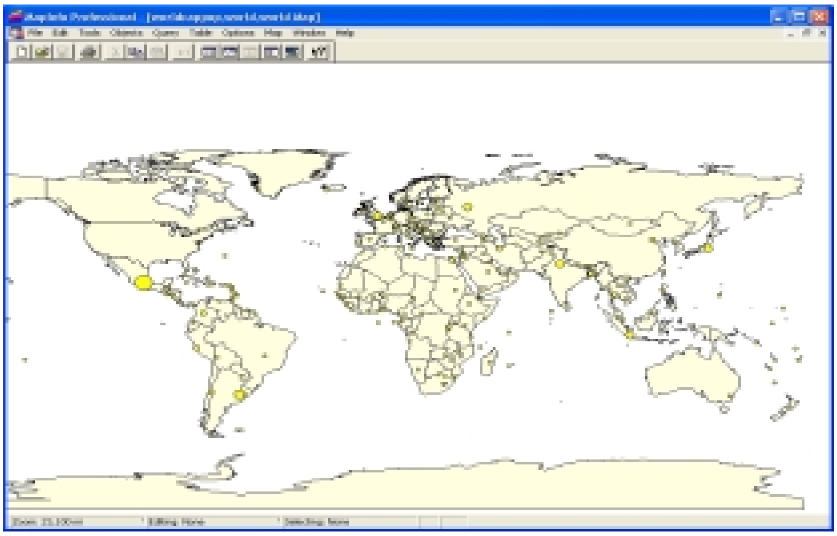

Figure 5: Example 4: Point Map (Numerical Data, Size Option)

6.5 Example 5 - Links With Numerical Data

Let us know give an example of a link-type map. For our particular example we use the region map consisting of US states as our background. We then use the file `linkdata.txt` from the `SampleData/` directory as our data. The file entries are:

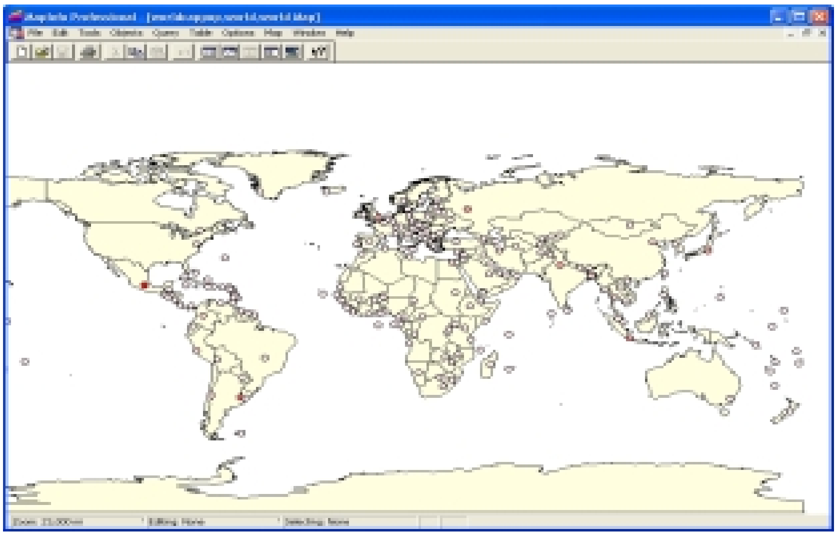

Figure 6: Example 4: Point Map (Numerical Data, Color Option)

```
 TX   NM   97.2
 NM   TX   54.8
JPN  AUS  -54.1
```

The first entry specifies that a flow of 97.2 units is passed from TX to NM. Similarly, 54.8 units are passed in the other direction. The command from the command prompt is:

> *gams link −−data=states −−datafile=SampleData/linkdata.txt −−tabfn=linkdata*

This creates a link file called *linkdata.tab* with positive flow going from IL to NY, DC to SD, and FL to DC. Negative flow goes from TX to WA, WA to TX, MA to NY, and SD to DC. The resulting map is shown in Figure 7.

The arrow widths are determined by the numeric values and the coloring by the sign of the numeric value. We remark that if flow goes in both directions for an origin-destination pair, then the GAMSMAP routines plot arrows in both directions. The coloring and arrow width is determined by the object with the maximum absolute numerical value. If bi-directional flow exists, then the choise of color and arrow width is specified in a status file called *status.log*.

7 ADVANCED FEATURES

In this section we will discuss some options to furher customize the way the maps created by GAMSMAP look. In particular, the user can customize the

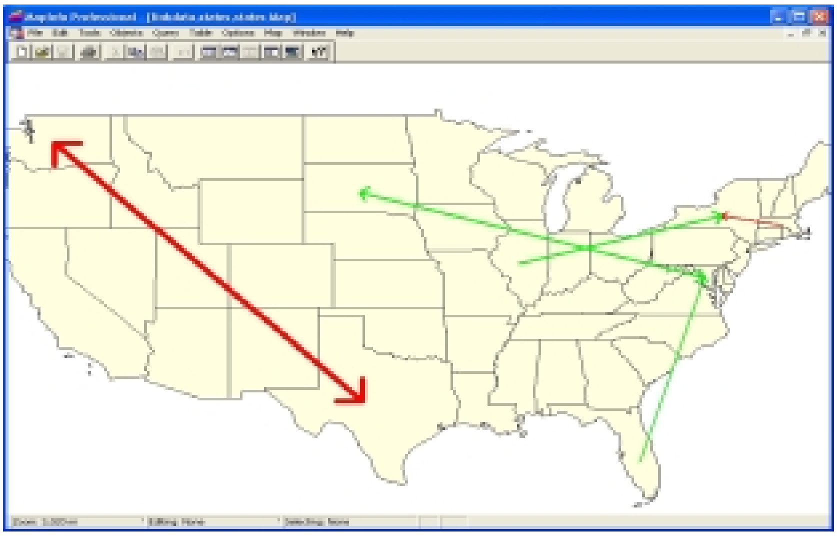

Figure 7: Example 5: Link Map (Numerical Data, Default Scaling)

numerical value scalings used for the region color shadings, the point sizes, and the link widths. Also, the user can specify a custom dBase script if a custom data extraction routine is desired.

7.1 Custom Scalings

The scalings for the numerical values for region shading, point size, and link width are specified in the default file *scaling.gms*. Values are computed by calculating the minimum and maximum absolute value of the numerical data, that is the range of the absolute value. The values are then assigned on a linear [0,1] scale with respect to 0 and the maximum.

If a different scaling is desired, the user can modify the file scaling.gms as needed. The option `−−scale=(filename).gms` then specifies the new scaling to be used. We shall give a brief description of the GAMS scaling code for each of the three main routines.

7.1.1 Region Color Shading Scale

Let us begin by looking at the GAMS code in scaling.gms for the region color shadings.

```
Set   steps /0*5/;
Table scaling(steps,*)
      lo         up      size
```

```
0     0.0          0.2     235
1     0.2          0.4     210
2     0.4          0.6     151
3     0.6          0.8     100
4     0.8          0.999    52
5     0.999        1.0       0;
```

The set *steps* specifies the number of different color shadings. The table *scaling* specifies the numeric shading value *size* if the scaled numeric value falls between *hi* and *lo*. The scaled numeric value is computed by linear interpolation so that the minimum corresponds to *lo=0* and that maximum to *hi=1*. The color shading value *size* can take on values between 0-255, where 255 corresponds to no shading (transparent) and 0 full coloring (solid).

7.1.2 Point Size Scale

The GAMS code in scaling.gms for the point sizes is:

```
Set   steps_s /0*5/;
Table scaling_s(steps_s,*)
       lo          up      size
0   -0.001       0.2       2
1    0.2         0.4       5
2    0.4         0.6       7
3    0.6         0.8       9
4    0.8         0.999    11
5    0.999       1.0      13;
```

The set *steps_s* specifies the number of different point sizes. The table *scaling_s* specifies the numeric point size value *size* if the scaled numeric value falls between *hi* and *lo*. The scaled numeric value is computed by linear interpolation so that the minimum corresponds to *lo=0* and that maximum to *hi=1*. The point size value *size* can take on values between 0-48, where 0 corresponds to no point at all and 48 to the largest point size.

7.1.3 Link Width Scale

For link.gms the GAMS code in scaling.gms is:

```
Set   steps_w /0*5/;
Table scaling_w(steps_w,*)
     lo            up      size
0  -0.001        0.2       2
```

```
1   0.2          0.4       3
2   0.4          0.6       4
3   0.6          0.8       5
4   0.8          0.999     7
5   0.999        1.0       7;
```

The set *steps_w* specifies the number of different link widths. The table *scaling_w* specifies the numeric link width value *size* if the scaled numeric value falls between *hi* and *lo*. The scaled numeric value is computed by linear interpolation so that the minimum corresponds to *lo=0* and that maximum to *hi=1*. The point size value *size* can take on values between 1-7, where 1 corresponds to the narrowest width and 7 to the largest width.

7.2 Using the --colorfile Option

Users can include a colorfile consisting of region element color assignments. This is useful, for example, if each region in a set should be assigned a different color according to some specified scheme. Users can include the file by specifying `--colorfile=(filename)`.

The `--colorfile` itself has the following format:

```
region  (IRGB value)
```

The IRGB value is a unique identifier combining the RGB values in a single integer. For example, 255 represents blue and 16711680 red. The respective IRGB value for each valid color is listed in the file `SampleData/rgbcolor.txt`. A sample colorfile entry is:

```
      TX       255
(or)  JPN 16711680
```

Sample colorfiles for US states and world countries are located in the `SampleData/` directory. They are called `states_col.txt` and `world_col.txt` respectively.

7.2.1 A --colorfile Example

Suppose we want to create a region map depicting US states, where each state is assigned a different color. We can use the `--colorfile` option to do this. We will use the sample US state colorfile located at `SampleData/state_col.txt`. The command is:

```
> gams region --data=states --colorfile=SampleData/states_col.txt
--tabfn=states
```

The resulting map is shown in Figure 8.

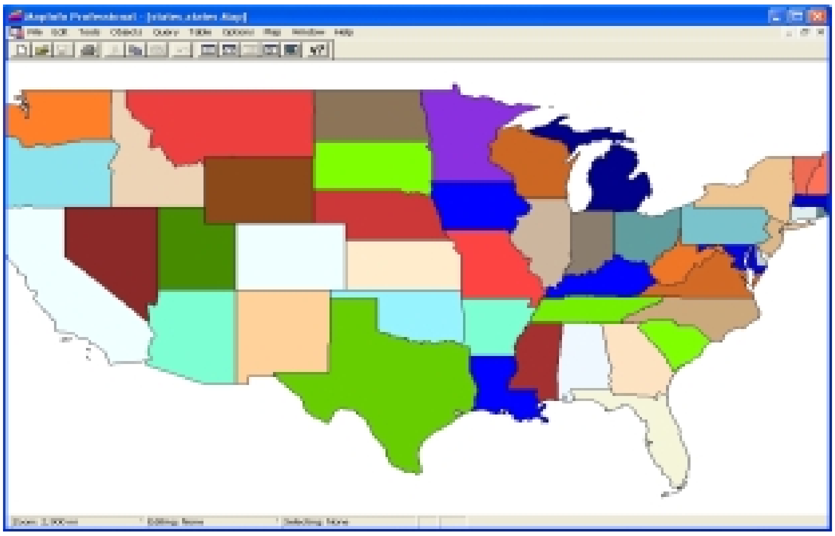

Figure 8: Example 5: Region Map (Colorfile option)

References

[1] A. Brooke, D. Kendrick, A. Meeraus, and R. Raman, *GAMS - A User's Guide*, GAMS Development Corporation, Washington, D.C., 1998.