GRTOOL
A Link-to-Grid Interface

by
Lukas and Heinz Spiess, EMME/2 Support Center

February 1994

Abstract:

GRTOOL is a set of macro and utility programs which provide an efficient link-to-grid interface in the context of the EMME/2 transportation planning software. Link based data, usually derived from network assignment results, can be transformed to a grid cell level, by splitting the link data according to the link's overlap with different grid cells, and accumulating it for each grid cell. Besides the program performing the link-to-grid conversion, GRTOOL also provides an EMME/2 macro for extracting link data, as well as a program to generate annotation files for displaying grid data within the framework of EMME/2 plots.

Heinz Spiess, EMME/2 Support Center

Sun Mar 3 22:21:06 MET 1996

Contents

Heinz Spiess, EMME/2 Support Center
Sun Mar 3 22:21:06 MET 1996

Introduction

EMME/2 provides a powerful environment for any kind of network computations. While the assignment modules are used to forecast the future use of the transportation infrastructure in terms of traffic volumes and speeds, the network calculator, module 2.41, can be used to postprocess the assignment results, computing whatever data that needs to be derived from the basic assignment results by simply entering the corresponding formulae. Typical examples of such postprocessing are the computation of level of service categories, emission of pollutants, noise levels or fuel and energy consumption. These computations are usually performed at the link level and the results are either summed or stored in link attributes for future use and graphic display on network plots.

In many situations, however, the aim of such models is not to obtain results at the network level, but the results are needed according to a different spatial aggregation. Two often used types of such aggregations are the summing up of values at the level of the traffic zones and the aggregation according to a *grid subdivision*, which cuts the area into rectangular cells of identical size. These kinds of aggregations cannot be implemented easily within the framework of the network calculator, in particular if one network link can cover more than one aggregation cell. Hence, more specialized tools are needed for this purpose.

GRTOOL is a general link-to-grid interface which can be used to convert any link based network data into a grid array with arbitrary cell dimensions. GRTOOL consists of the following components:

grmacro:
An EMME/2 macro which takes as parameters a set of link attributes or expressions that correspond to the variable to be converted to the grid level. Using module 2.41, it generates an output file containing, for each link, the coordinates of the I- and J-node, as well as the values of the specified variables.

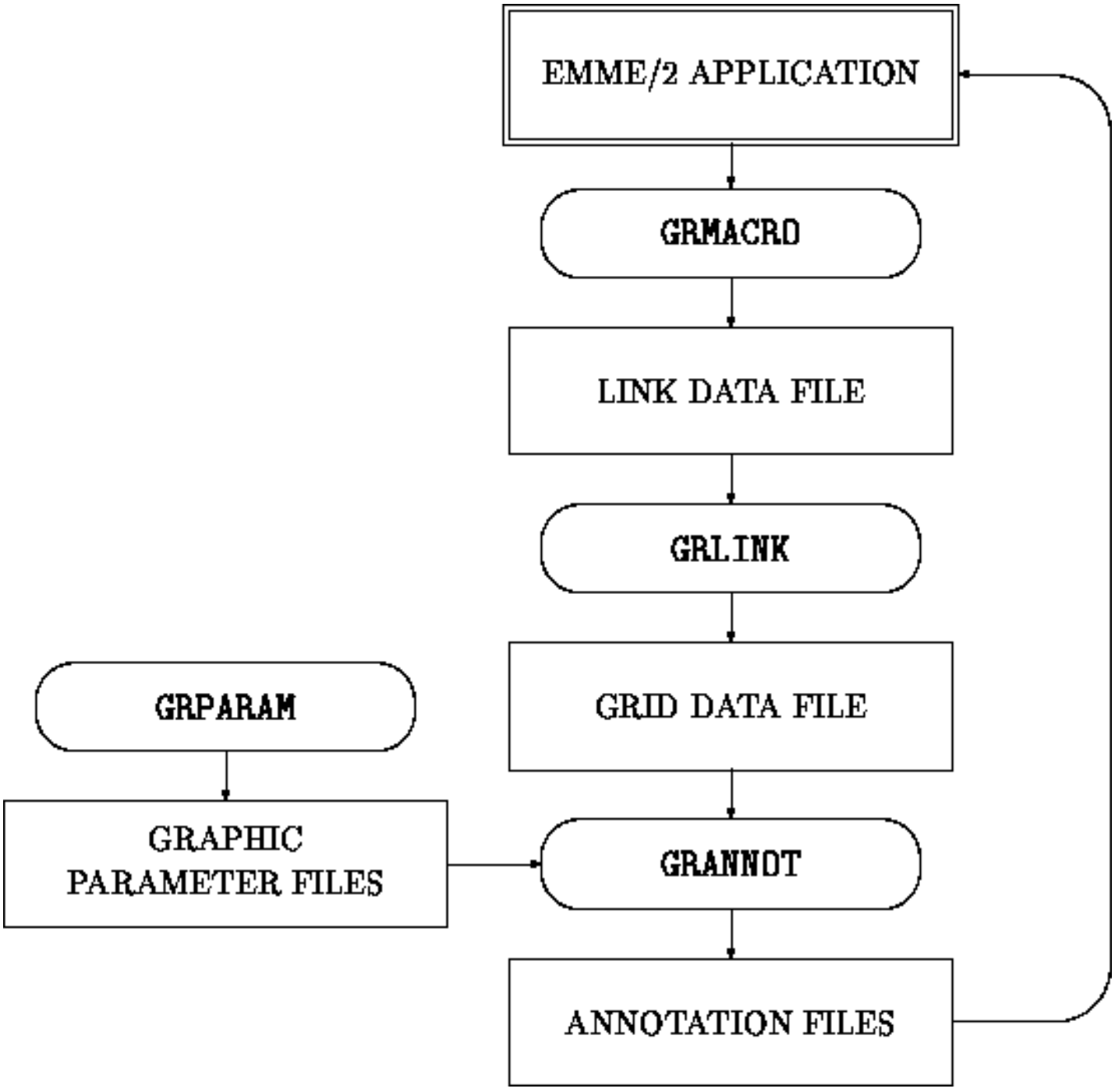
grlink:
A program written in AWK which performs the actual link-to-grid conversion. Given the size of the grid cell, it reads a link based file (usually generated with `grmacro`) and outputs a grid based result file, which contains the same variables as the link file, but accumulated for each grid cell.

grannot:
A program written in AWK which reads a grid file generated by `grlink`, extracts one value for each grid cell from it and generates an annotation file which can be displayed in EMME/2, either alone or superimposed on a network plot. The type of display is controlled by a parameter file which is generated with the `grparam` utility.

grparam:
A utility program, written in AWK, used to define the layout and display parameters (such as intervals, colors and hatching for the cell categories, titles, position and texts for the legend, ...). To do this, the user is guided through an ``EMME/2-like'' dialog. The so defined set of parameters is

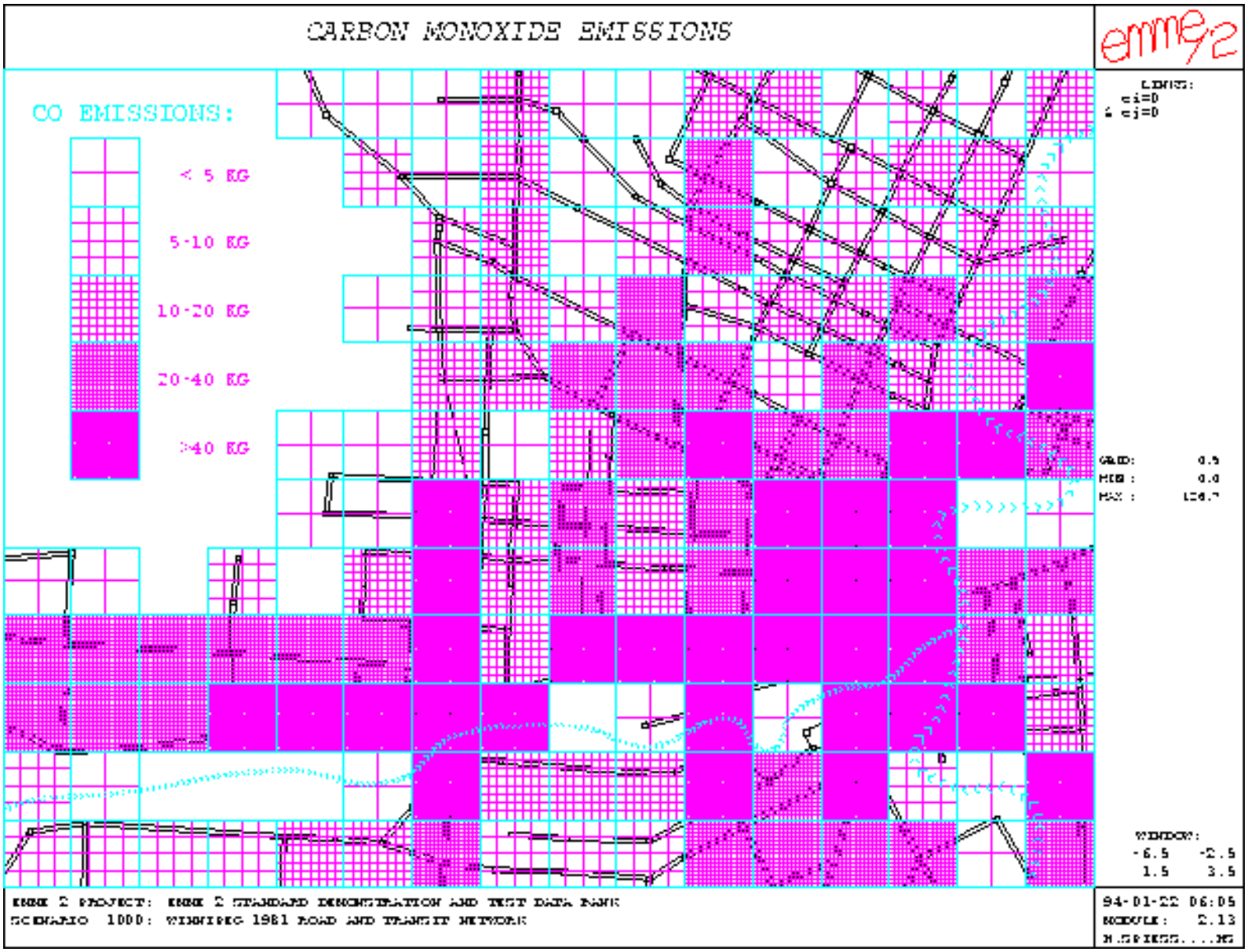
saved in a file and used by `grannot` for producing displays of this particular kind.

The diagram below shows the interaction between the various components of GRTOOL and the data files accessed by each:



A typical application of GRTOOL is in the area of environmental impact studies, where often the emissions of the various transportation related pollutants (such as CO, HC, NO, SO_2) have to be computed at the link level, and afterwards are converted to the grid level. These may then be combined with the pollutants generated by the stationary sources (industry, heating) in order to display them as emission maps. In this case, the network calculator is first used to compute extra link attributes containing the emissions for all four pollutants. Then `grmacro` is used to generate a link file with all four pollutant emissions. This file is

converted into a grid cell file by `grlink`, which can be accessed by `grannot` to produce plots of the individual pollutants or combinations thereof. The graphic representations can be in the form of differently colored and hatched interval classes, as proportional squares, or as numeric values displayed in the grid cells. The example below shows the varying levels of carbon monoxide emissions, displayed as squares with different hatchings:



GRTOOL has been developed at the EMME/2 Support Center by Lukas and Heinz Spiess. DOS and UNIX versions of GRTOOL are made available to EMME/2 users at no charge, as a ``copylefted'' user contribution. Thus, GRTOOL is *not* part of the commercial EMME/2 software distribution and therefore its availability implies no obligations, on the part of INRO Consultants. The GRTOOL software is distributed *as is* and there is no warranty for this free software. You are allowed to copy the unmodified complete version of GRTOOL and pass it along to other EMME/2 users for free, but you may not sell it for profit. You can make changes to GRTOOL provided that a) you do not remove the copyright notices, b) you clearly

identify the changes with at least the date and your name, and c) you rename the modified files.

Heinz Spiess, EMME/2 Support Center
Sun Mar 3 22:21:06 MET 1996

Link-to-Grid Conversion

In this section we briefly describe the basic terms and concepts used in the link-to-grid interface implemented in GRTOOL.

A *link* is considered a straight line connection between two network nodes. For each link *a*, the coordinates of the I-node (XI_a, YI_a) and of the J-node (XJ_a, YJ_a) are known from the network definition. A *width* w_a is associated with each link. It is specified in the same units as the network coordinates and serves to define the geometric link representation, which is a centered link bar of width w_a extending between I- and J-node. For each link a set of data values v_a^1, v_a^2, v_a^2, ..., represents the data that is to be transformed to the grid level.

A *grid* is a subdivision of the network plane by equally spaced horizontal and vertical parallel lines. The distance between two neighboring parallel lines is called the *grid distance*. It is defined in the same units as the network coordinates. The relative position of the parallels is such that a horizontal and a vertical line always pass through the origin of the coordinate system, i.e. point *(0,0)*. The resulting square shaped subdivisions are called *grid cells*. They are identified by their lower left corner coordinates.

Each link value v_a^k is split proportionally to the area overlap with the corresponding grid cells. The resulting partial values are then accumulated for each grid cell. The fact that the link is represented as a two-dimensional bar, instead of a one-dimensional line segment, avoids any ambiguities that might otherwise arise when a link coincides with a cell boundary or passes through a cell corner. This is illustrated in the following figure, which shows how the data for a given link (shown in solid black) is distributed to the neighboring grid cells: Since, depending on the network size and the chosen grid distance, the link-to-grid conversion can be quite a complex task, an efficient implementation is very important. The following is an outline of the method implemented in `grlink`:

1. The link data file is read and, for each link, the coordinates of the corresponding link bar are computed. Using a recursive approach, the bar is intersected with all intervening grid cells and the intersection areas are computed. For each grid cell that intersects with the link bar a record is written to a temporary file containing the grid cell coordinates and the corresponding shares of the data values v_a^k.
2. The temporary file containing the link specific grid records is sorted according to the grid cell coordinates, resulting in another temporary file in which all data concerning one grid cell is now

contiguous.

3. The sorted temporary file is read sequentially. For each grid cell, the corresponding data values are accumulated and the totals are written into the grid data file.

Heinz Spiess, EMME/2 Support Center
Sun Mar 3 22:21:06 MET 1996

Installation of GRTOOL

GRTOOL is available for UNIX and MS-DOS systems. These versions are distributed on MS-DOS diskettes. For the installation of GRTOOL on your system, follow the instructions for either MS-DOS or UNIX given below.

Heinz Spiess, EMME/2 Support Center
Sun Mar 3 22:21:06 MET 1996

Installation on MS-DOS Systems

The MS-DOS distribution of GRTOOL contains the following files:

File:		*Description:*
`GRMACRO.MAC`	*	EMME/2 macro used to produce link data file
`GRLINK.BAT`	*	Batch file for transforming link-to-grid conversion
`GRPARAM.BAT`	*	Batch file for creating a graphic parameter file
`GRANNOT.BAT`	*	Batch file for creating an annotation from a grid file
`GRLINKA.EXE`	*	Binary program called by `GRLINK.BAT`
`GRPARAMA.EXE`	*	Binary program called by `GRPARAM.BAT`
`GRANNOTA.EXE`	*	Binary program called by `GRANNOT.BAT`
`GSORT.EXE`	*	GNU sort utility, distributed under the terms of the GNU General Public Licence, used by `GRLINK`.
`GAWK.EXE`		GNU AWK interpreter, distributed under the terms of the GNU General Public Licence
`GRLINKA.AWK`		AWK source code of `GRLINKA.EXE`
`GRPARAMA.AWK`		AWK source code of `GRPARAMA.EXE`
`GRANNOTA.AWK`		AWK source code of `GRANNOTA.EXE`
`README`		Text file with last minute informations which may not be contained in the printed documentation

To install GRTOOL, create a new directory `\GRTOOL` and copy all files marked with * in the above list into it. Add this new directory to the `PATH=` statement in your `AUTOEXEC.BAT` file. Copy the macro file `GRMACRO.MAC` into the EMME/2 application directory or into one of the directories that appear in your E2MACROS variable, or, alternately, append the term ``;\GRTOOL\?.MAC'' to your E2MACROS variable.

The GRTOOL programs check the path directories to find the location of the GRTOOL program files. Due to limitations of the DOS batch programming language, this auto detection might fail for very long settings of the PATH variable. In this case, it is possible to add a statement of the form ``GRTOOL=C:

\GRTOOL" to AUTOEXEC.BAT, to indicate explicitly the location of the GRTOOL files, thereby disactivating the autodetection feature.

The AWK source code files are not needed to run the GRTOOL programs. They are provided for your information. Note that if the compiled binaries of these AWK programs are not found, the GRTOOL programs will call the GAKW interpreter with the corresponding source files instead.

Next Up Previous Contents

Next: Installation on UNIX Systems **Up:** Installation of GRTOOL **Previous:** Installation of GRTOOL

Heinz Spiess, EMME/2 Support Center
Sun Mar 3 22:21:06 MET 1996

Installation on UNIX Systems

The UNIX distribution of GRTOOL contains the following files:

File:	*Description:*
`grmacro.mac`	EMME/2 macro used to produce link data file
`grlink`	Shell script for performing link-to-grid conversion
`grparam`	Shell script for creating a graphic parameter file
`grannot`	Shell script for creating an annotation from a grid file
`grlinka.awk`	AWK program called by `grlink`
`grparama.awk`	AWK program called by `grparam`
`grannota.awk`	AWK program called by `grannot`
`readme`	Text file with last minute informations which may not be contained in the printed documentation

Create a new directory ``grtool'' and copy the GRTOOL distribution files into it and add this directory to your command search path. Copy the macro file `grmacro.mac` into your EMME/2 application directory or into one of the directories that appears in your E2MACROS variable, or, alternately, append the term ``/.../grtool/?.mac'' to your E2MACROS variable.

The GRTOOL programs check the path directories to automatically find the location of corresponding AWK files. This autodetection may be disactivated by setting an environment variable GRTOOL to the directory in which the GRTOOL files are stored.

By default the GRTOOL programs call the AWK interpreter by the name `gawk` (which is the name used by the freely distributed Gnu AWK). A different AWK interpreter can be used by defining an alternate name (such as ``awk'' or ``nawk'') in the environment variable GRTOOLAWK. Note, however, that GRTOOL requires a so-called ``new version'' of AWK, as described by Aho, Kerninghan and Weinberger in *The AWK Programming Language*, Addison-Wesley, 1988.

Heinz Spiess, EMME/2 Support Center
Sun Mar 3 22:21:06 MET 1996

Working with GRTOOL

In this section each of the GRTOOL programs is explained in detail. The functionality and the calling conventions for all the programs are basically the same for both MS-DOS and UNIX versions of GRTOOL.

Heinz Spiess, EMME/2 Support Center
Sun Mar 3 22:21:06 MET 1996

GRMACRO - EMME/2 Macro to Generate a Link Data File

GRMACRO is an EMME/2 macro which is used to generate a link data file, containing attribute values which later are to be converted to a grid subdivision by means of program GRLINK. The macro is called either from the main menu or from the primary select of module 2.41, using the following calling sequence:

~<grmacro *width var 1 [var 2] [var 3] [var 4]*

The parameter *width* corresponds to the link width (in coordinate units) used for the link-to-grid conversion. It can be either a constant (e.g. ``0.01'') or any network expression (e.g. ``lanes*.005'').

Following the link width, up to four link attributes can be specified which are to be written to the link file.

If, e.g., .01 (=10 meters, if coordinates are given in kilometers) is assumed as link width and the emissions for carbon monoxide, nitrous oxides and hydrocarbons have already been computed by the corresponding formulae and saved in the extra link attributes @co, @nox and @hc, the macro would be called as follows

```
~<grmacro 0.01 @co @nox @hc
```

This macro will generate a batch output file which can be read directly by GRLINK. Note that the name of the output file depends on the current settings of the flexible file naming, which is set by the menu command batchout=. It is therefore up to the user to set the output file name correctly, and make sure that the output file is empty, before calling GRMACRO. Due to the nature of the output generated by module 2.41, the link data file will contain, in addition to the specified link attributes, also the total of all attributes, labeled ``result''.

The dialog for the selection of the subset of links is not answered directly by the macro, but is passed through to the terminal. Thus, the set of links to be processed can be chosen freely by the user.

If the macro is started at the main menu, the user is prompted to enter the name of the link data file to be generated. Note that this file must not already exist or it must be empty, since otherwise the link data information is appended to the end of the file.

As mentioned above, it is possible to start GRMACRO directly from the primary select of module 2.41.

This makes it possible to compute link attributes into temporary link attributes `tmpln` (which are lost when module 2.41 is left) and call GRMACRO directly afterwards. If doing so, please note that `tmpl9` is used by GRMACRO to store the link widths. Thus, it should not be used as one of the calling parameters. If the macro is started from module 2.41, the output is sent to the batch output file, as defined by the current setting of the `batchout=` menu command.

Heinz Spiess, EMME/2 Support Center
Sun Mar 3 22:21:06 MET 1996

GRLINK - Transform Link Data to Grid Data

GRLINK is the program which implements the link-to-grid conversion, i.e. it transforms a link data file into the corresponding grid data file. The following calling sequence is used:

`grlink` *input-file grid-distance [output-file]*

The parameter *input-file* specifies an existing link data file which is to be read and transformed into a grid data file. Usually, the link data file is created by GRMACRO, but files from other sources can be used as well, as long as the file is formatted according to the format specifications given in section 5.1. In particular, it is important to note that the link file can contain more than five data values per link, even if the files generated by GRMACRO are limited to a maximum of four values plus the total.

The parameter *grid-distance* specifies the grid distance, in network coordinate units, to be used for the conversion. By default, the same grid distance is used in horizontal and vertical direction, i.e. a square grid is assumed. However, it is also possible to define a rectangular grid by specifying the horizontal and vertical grid distances individually, separated by a slash (e.g. ``.5/.25").

The resulting grid data file is written to the file specified in the *output-file* parameter. If this parameter is not specified, the output is sent to the standard output for UNIX, to the console device on MS-DOS. The format of the resulting grid data file is described in detail in section 5.2.

Note that, depending on the size of the network and the chosen grid distance, as well as the performance of the system used, running GRLINK might be quite time consuming.

Heinz Spiess, EMME/2 Support Center
Sun Mar 3 22:21:06 MET 1996

GRPARAM - Define a Graphic Parameter File

GRPARAM is a utility program which is used to set up a *graphic parameter files*, i.e. file which specifies the details how to produce one particular kind of grid display. Once such a file has been created it can be used to produce any number of annotation files of this kind.

GRPARAM is called as follows:

```
grparam
```

GRPARAM generates a dialog that looks very similar to a standard EMME/2 dialog. Thus, for someone who is used to work with EMME/2, it is easy to operate GRPARAM. Note, however, that for Yes/No and Select type questions, unlike the EMME/2 dialog, it is not possible to answer with ``r'' or ``q'' for restart or quit.

In the dialog the user is prompted to specify the name of the parameter file that is to be generated. Also, he has the possibility to specify an existing parameter file as default file, which is useful to modify a parameter file or to create a variant of it.

GRPARAM distinguishes three main types of grid representation:

Classes:
The grid cell values are divided in different classes. For each class a lower and upper bound defines the interval of grid cell values that belong to this class. All grid cells belonging to the same class are displayed using the same color and hatching parameters. Horizontal, vertical and cross-hatching is available.

Numeric values:
The grid cell values are displayed numerically in the grid cell. The format, size, color and position of the numbers within the grid cells can be chosen freely.

Proportional hatching:
The grid cells are hatched vertically with the number of hatch lines increasing proportionally to the grid value. Color and scale factor are specified by the user.

The format of the generated output file is documented in section 5.3.

Heinz Spiess, EMME/2 Support Center
Sun Mar 3 22:21:06 MET 1996

GRANNOT - Generate Annotation from Grid Data File

GRANNOT is a program which extracts a set of values from a grid data file (usually produced with GRLINK) and generates an EMME/2 annotation file. This annotation file can be displayed in EMME/2, either superimposed on a network plot, or in its own right.

The details of the graphic representation are taken from a graphic parameter file, which contains details as to the grid attribute to be displayed, the type of representation, the class definitions, colors, hatching, text size and location, as well as contents and position of the legend.

GRANNOT is called as follows:

`grannot` *[-d item] parameter-file input-file [annotation-file]*

The parameter *parameter-file* specifies the name of the graphic parameter file which contains the details of desired graphic representation. This file is usually created with GRPARAM.

The *input-file* parameter corresponds to the name of the grid data file containing the values to be displayed. This file is usually generated with GRLINK, but files from other sources can be used as well, as long as their format correspond to the specifications given in 5.2.

The optional parameter *annotation-file* can be used to specify the name of the resulting annotation file. If it is omitted, the output is sent to the standard output device on UNIX and to the console device on MS-DOS.

By default the data item to be extracted from the grid file is specified in the parameter file. With the option `-d` *data* this default can be overwritten by specifying the name or number of an alternate variable which is to be accessed in the grid file.

One call to GRANNOT will produce an annotation representing one grid attribute. If the grid file contains several grid attributes --which is usually the case--, GRANNOT is called repeatedly with the same grid data file but different parameter files.

Heinz Spiess, EMME/2 Support Center
Sun Mar 3 22:21:06 MET 1996

GRTOOL File Formats

This section contains the detailed file format specification of the various data and parameter file types that are used by GRTOOL. For normal operation where all files are generated by GRTOOL itself, a user does not need to be concerned about these details.

However, when the GRTOOL programs are used with data files that are created outside the GRTOOL / EMME/2 context, it is necessary to make sure that the file used corresponds to the proper file format rules.

Unless explicitly stated otherwise, free format is used to interpret the contents of the individual records. Records are composed of several fields which are separated by blank space.

Heinz Spiess, EMME/2 Support Center
Sun Mar 3 22:21:06 MET 1996

Link Data File

A link data file is a standard ASCII text file which, for each applicable network link, contains one record with the following information:

Field:	*Name*	*Contents:*
1	`inode`	Number of I-node of link (not explicitly used by GRTOOL)
2	`jnode`	Number of J-node of link (not explicitly used by GRTOOL)
3	`xi`	X-coordinate of I-node of link
4	`yi`	Y-coordinate of I-node of link
5	`xj`	X-coordinate of J-node of link
6	`yj`	Y-coordinate of J-node of link
7	*width*	link width in the same units as used for network coordinates
8	*var1*	First data value for link (v_a^1)
9-...	*vark*	Second, third, ...data values for link ($v_a^k, k = 2, \ldots$)

The link records can appear in any order, i.e. sorting of the links is not required.

Optionally the link records can be preceded by a *label record* of the form:
`inode jnode xi yi xj yi` *width var1 var2 var3 ...*
If present, this record allows to identify the data values v_a^k by the names *var1*, *var2*, *var3*, ...

Any record in a link data file which starts with the character # is considered as a comment. Comments may appear anywhere in the file, but they are simply ignored.

Usually, link data files are generated by macro GRMACRO and are processed by program GRLINK. Even though GRMACRO generates link data files with up to 5 data values only, the format of the link data file itself does not impose an explicit maximum for the number of data values.

Heinz Spiess, EMME/2 Support Center
Sun Mar 3 22:21:06 MET 1996

Grid Data File

A grid data file is a standard ASCII text file which contains for each applicable grid cell one record with the following information:

Field:	*Name*	*Contents:*
1	xg	X-coordinate of lower left corner of grid cell
2	yg	Y-coordinate of lower left corner of grid cell
3	*var1*	First data value for grid cell
4-...	*var*k	Second, third, ...data values for grid cell

The grid records can appear in any order, i.e. sorting of the grid cell records is not required.

Optionally the grid records can be preceded by a *label record* of the form:
xg=*distx* yg=*disty var1 var2 var3 ...*

If present, this record allows to identify the data values v_a^k by the names *var1, var2, var3, ...*

Any record in a grid data file which starts with the character # is considered as a comment. Comments may appear anywhere in the file, but they are simply ignored.

Grid data files are usually generated by program GRLINK and are read by program GRANNOT in order to generate grid displays in to the form of annotation file.

Heinz Spiess, EMME/2 Support Center
Sun Mar 3 22:21:06 MET 1996

Graphic Parameter File

A graphic parameter file contains all the parameters needed by GRANNOT to extract a data set from a grid file and generate an annotation file from it. It includes information on grid distance, variable to access, type of representation, colors, hatch widths, title, legend, etc.

Each record of a parameter file starts with a two letter keyword indicating the type of record. Depending of the type of record, the keyword is followed by one or more type specific parameter values and optional comments. The parameter values are separated by blanks. Texts, such as title and legend strings, are surrounded by single quotes.

Records which start with the character # are considered as comments. They may appear anywhere in the file.

The following tables list the various record types and the associated parameters. The first table shows the record types which are applicable to all types of representation:

Record	*Parameter*		*Contents*
Type	*No.*	*Type*	*Description*
GD	1,2	real	Horizontal, vertical grid distances
GC	1	int	Color index used for grid contour
CG	1	0/1	Indicator flag for drawing complete grid
DN	1	string	Name or position of grid file variable to be accessed
RP	1	int	Type of representation: 0=classes, 1=numeric values, 2=proportional hatching

If a representation by classes is chosen, the following record types are used to define the intervals and the legend type and layout:

Record	*Parameter*		*Contents*
Type	*No.*	*Type*	*Description*
LG	1	int	Type of legend: 0=none, 1=standard, 2=customized

LB	1,2	real	Width, height of legend boxes
LK	1,2	real	X-, Y-coordinate of lower left corner of first legend box
TD	1,2	real	Relative X-, Y-offsets for legend text
TS	1	real	Size of legend text
LD	1,2	real	Relative X-, Y-offsets for next legend box
LT	1	text	Legend title (surrounded by single quotes)
TI	1	real	Size of legend title
TK	1,2	real	X-, Y-offsets for legend title from lower left corner of first legend box
NC	1	int	Number of interval classes
LC	1	int	Class number to be defined in this record
	2,3	real	Lower, upper threshold defining the value interval for this class. The interval is defined including the lower, but excluding the upper threshold value.
	4	int	Color used for hatching grid cells belonging to this class
	5,6	real	Horizontal, vertical hatching distance for this class (0 indicating no hatching in this direction)
	7	text	Legend text for this class surrounded by single quotes

For representation by display of numerical values within the grid boxes the following parameter records are used:

Record	*Parameter*		*Contents*
Type	*No.*	*Type*	*Description*
PC	1	int	Color used to display the numeric values
NS	1	real	Text size used to display the numeric values (default *1/8* of grid distance)
DT	1	int	Minimum field width used for numeric value (default 5)
DF	1	int	Number of digits after the decimal point (default 0)
DG	1,2	real	X-, Y-offset for numeric value from lower left corner of grid box (default 10% of horizontal, vertical grid distance)

Finally, if the grid variable is displayed using proportional hatching, the following parameter records are used:

Record	*Parameter*		*Contents*
Type	*No.*	*Type*	*Description*
PC	1	int	Color used for proportional hatching
SC	1	real	Scale factor, the number of hatch lines for a grid cell is obtained by dividing the corresponding attribute value by this scale factor.

Heinz Spiess, EMME/2 Support Center
Sun Mar 3 22:21:06 MET 1996

Annotation File

Annotation files containing graphical grid representations are generated by program GRANNOT. These files can be read and displayed by a graphic module of EMME/2. For a detailed description of the format of EMME/2 annotation files, please refer to the EMME/2 User's Manual.

Heinz Spiess, EMME/2 Support Center
Sun Mar 3 22:21:06 MET 1996

An Example

In the following detailed example, we show a typical application of GRTOOL using the standard EMME/2 demonstration data bank with the Winnipeg network.

First, the emissions for carbon monoxide, hydrocarbons and nitrous oxides are computed using the network calculator, module 2.41, and the resulting emissions are saved in the link extra attributes @co, @hc and @nox. For the sake of this example, functions were used which are based on the values published by the Swiss Federal Office for the Protection of the Environment for private vehicles for the year 1984. . These functions are fourth degree polynomials of the travel speed in km/h on the link which yield the amount of pollutants generated by one vehicle in g/km.

$$CO = 38.10954 - 31.94919\frac{v}{34} + 9.77821\left(\frac{v}{34}\right)^2 - 1.22547\left(\frac{v}{34}\right)^3 + .099575\left(\frac{v}{34}\right)^4$$

$$HC = 4.768756 - 4.846669\frac{v}{34} + 2.16595\left(\frac{v}{34}\right)^2 - 0.42800\left(\frac{v}{34}\right)^3 + .032168\left(\frac{v}{34}\right)^4$$

$$NO_x = 1.289830 + 1.646201\frac{v}{34} - 1.95990\left(\frac{v}{34}\right)^2 + 0.95789\left(\frac{v}{34}\right)^3 - .129221\left(\frac{v}{34}\right)^4$$

Since distances in the Winnipeg data bank are given miles, care must be taken regarding the units when applying the above formulae.

Once the emissions are computed at the link level and saved in the corresponding extra attributes, macro GRMACRO is called from the main menu as follows

```
~<grmacro .01 @co @hc @nox
```

When GRMACRO is running, the user is prompted to specify, in the usual manner, the subset of links to include in the link data file. Also, since the macro is started from the main menu, the user is asked to specify the name of the output file to be generated. The following shows the screen output generated in this example by GRMACRO:

```
Enter: Next module=~<grmacro .01 @co @hc @nox
********************** GRMACRO (2.1) ****************************
GRMACRO - macro to generate link data file for use with GRLINK

Usage: ~<grmacro <width> <val1> <val2> <val3> ...

where   <width>  constant or link attribute containing link width
        <valK>   link attribute containing link data K

Notes: - macro is started from main menu or primary select of 2.41
       - tmpl9 is used to store link widths
       - link subset and name of output file is asked interactively
       - switch 22 should be off in order to avoid eng. notation
*****************************************************************

result    =   0 * (xi + yi + xj + yj + tmpl9) + @co + @hc + @nox
```

```
Enter: Selected link types or attributes (from, to)
= >>ci=0
= >>and cj=0
= >>
Enter: Name of link data file to be generated= >>emission.lnk

Macro ``~<grmacro .01 @co @hc @nox'' terminated normally.
*****************************************************************
Enter: Next module=
```

The generated link data file is called `emission.lnk` and contains the coordinates, widths and specified link attributes for the selected links, as shown below:

inode	jnode	xi	yi	xj	yj	tmpl9	@co	@hc	@nox	result
165	166	.773	.848	.903	.961	.01	.69519	.09486	.1961	.98615
165	1055	.773	.848	.61	.669	.01	11.6673	1.33537	1.46016	14.4629
166	165	.903	.961	.773	.848	.01	5.07276	.5806	.63485	6.28821
166	167	.903	.961	.951	1.042	.01	.62567	.08537	.17649	.88754
167	166	.951	1.042	.903	.961	.01	4.56548	.52254	.57137	5.65939
167	168	.951	1.042	1	1.107	.01	.41711	.05692	.11766	.59169
.....										
1064	1063	.578	-.305	.432	-.289	.01	12.3012	1.42774	.8722	14.6011
1065	477	-3.643	-9.56	-1.744	-9.349	.01	.29588	.04677	.12603	.46867
1065	546	-3.643	-9.56	-8.79	-9.089	.01	.8352	.13201	.35575	1.32297
1066	727	-2.059	.674	-2.038	.969	.01	3.84376	.44217	.51005	4.79599
1067	493	-2.02	-4.056	-2.004	-4.299	.01	4.90105	.60934	1.00488	6.51527
1067	494	-2.02	-4.056	-2.004	-3.634	.01	1.55136	.2348	.58801	2.37417

This link file `emission.lnk` is now converted to a grid with a cell size of 0.5 x 0.5 miles. This is achieved by calling the program GRLINK as follows:

```
grlink emission.lnk 0.5 emission.grd
```

The resulting grid data file `emission.grd` contains the grid cell values of same variables as contained in the link file.

```
# Created by GRLINK (2.1)
  xg=0.5   yg=0.5      @co      @hc     @nox   result
   0.000    0.000   41.108    4.755    3.980   49.843
   0.000    0.500   69.116    7.955    8.200   85.271
   0.000    1.000   19.950    2.367    3.142   25.459
   0.000    1.500   28.703    3.358    4.063   36.124
   0.000    2.000    6.053    0.690    0.702    7.444
   0.000    2.500   13.790    1.584    1.523   16.897
   0.000    3.000    7.207    0.833    0.570    8.610
   0.000    3.500    0.506    0.058    0.066    0.630
   0.000    4.000    1.593    0.183    0.212    1.989
   0.000    4.500    0.023    0.004    0.009    0.036
   0.000    5.000    6.533    0.837    1.503    8.873
   0.000    5.500    0.179    0.023    0.041    0.242
   0.000    7.000    0.514    0.081    0.219    0.814
   0.000   -0.500   81.059    9.301    7.771   98.130
   0.000   -1.000  107.999   12.425    9.706  130.130
   0.000   -1.500   49.093    5.690    4.383   59.166
   0.000   -2.000    8.431    0.977    0.667   10.074
   0.000   -2.500   23.692    2.731    1.988   28.411
......
```

The emission values contained in the file `emission.grd` can now be displayed in the form of annotations which are generated with program GRANNOT. But in order to run GRANNOT, it is first necessary to prepare the corresponding graphic parameter files. This is best done using the program GRPARAM.

The first example of the generation of such a parameter file is for generating a display of the type shown before,

dividing the grid cells into 5 different classes according to the emission levels:

```
Enter: Name of existing parameter-file for default values (optional)=
Enter: Name of parameter-file to be generated = co.par
Enter: Horizontal[,vertical] grid distances (1,1) = .5 .5
Draw complete grid (n) ? n
Enter: Grid color (4) = 4
Enter: Name or number of data item to be displayed (1) = @co
Select: Type of gird representation
        1= Classes defined by value interval
        2= Numeric values
        3= Hatching with proportional density
       (1) 1
Select: Type of legend
        1= None
        2= Default
        3= Customized
       (1) 2
Enter: Text for legend title (*) = CO:
Enter: Lower left corner of first legend box (xl, yl) (*,*) = 6.5,5
Enter: Number of classes (0) = 5

Enter: Class interval  1: (from, to) (0, 0) =0 5
Enter: Color for class  1 (0) = 5
Select: Hatching pattern for class  1
        1= Horizontal lines
        2= Vertical lines
        3= Horizontal and vertical lines
       () 3
Enter: Distance between 2 horizontal lines (0) = .25
Enter: Distance between 2 vertical lines (0) = .25
Enter: Text for class  1 () =   < 5 KG

Enter: Class interval  2: (from, to) (0, 0) =5 10
Enter: Color for class  2 (0) = 3
Select: Hatching pattern for class  2
        1= Horizontal lines
        2= Vertical lines
        3= Horizontal and vertical lines
       () 3
Enter: Distance between 2 horizontal lines (0) = .125
Enter: Distance between 2 vertical lines (0) = .125
Enter: Text for class  2 () =  5-10 KG

Enter: Class interval  3: (from, to) (0, 0) =10 20
Enter: Color for class  3 (0) = 7
Select: Hatching pattern for class  3
        1= Horizontal lines
        2= Vertical lines
        3= Horizontal and vertical lines
       () 3
Enter: Distance between 2 horizontal lines (0) = .0625
```

```
Enter: Distance between 2 vertical lines (0) = .0625
Enter: Text for class  3 () = 10-20 KG

Enter: Class interval  4: (from, to) (0, 0) =20 40
Enter: Color for class  4 (0) = 8
Select: Hatching pattern for class  4
        1= Horizontal lines
        2= Vertical lines
        3= Horizontal and vertical lines
       () 3
Enter: Distance between 2 horizontal lines (0) = .03125
Enter: Distance between 2 vertical lines (0) = .03125
Enter: Text for class  4 () = 20-40 KG

Enter: Class interval  5: (from, to) (0, 0) =40,999999
Enter: Color for class  5 (0) = 2
Select: Hatching pattern for class  5
        1= Horizontal lines
        2= Vertical lines
        3= Horizontal and vertical lines
       () 3
Enter: Distance between 2 horizontal lines (0) = .015625
Enter: Distance between 2 vertical lines (0) = .015625
Enter: Text for class  5 () =   >40 KG
```

The result of the above executing of GRPARAM is a graphic parameter file `co.par` with the following contents:

```
GD         .5         .5 Horizontal,vertical grid distances
CG          0            Draw complete grid
GC          4            Grid color
DN        @co            Name or number of data item to be displayed
RP          0            Representation (0=Class,1=Numbers,2=Proportional)
LG          1            Legend (0=None,1=Default,2=Customized)
LT 'CO:'                 Legend title
LK        6.5          5 Lower left corner of first legend box (xl, yl)
NC          5            Number of classes
LC  1         0         5   5    .25    .25 '  < 5 KG'
LC  2         5        10   3   .125   .125 ' 5-10 KG'
LC  3        10        20   7  .0625  .0625 '10-20 KG'
LC  4        20        40   8  .03125  .03125 '20-40 KG'
LC  5        40    999999   6  .015625  .015625 '  >40 KG'
```

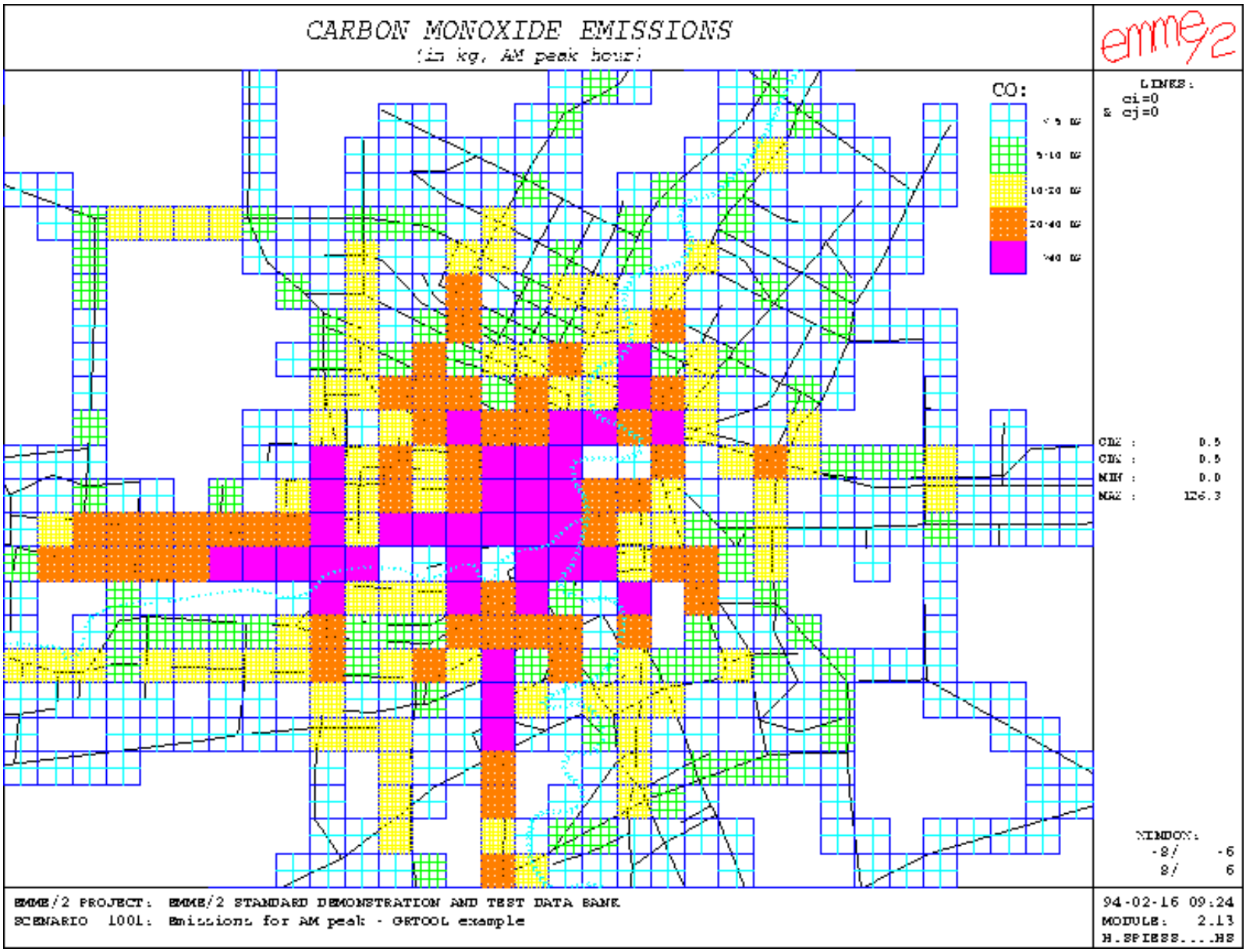


Figure 1: Example of grid representation by interval classes

Using the resulting graphic parameter file `co.par` the program GRANNOT is now used to produce an annotation file `annotc` using the command

```
grannot co.par emission.grd annotc
```

which can be superimposed on an EMME/2 network plot to obtain the map of carbon monoxide emission levels shown in in Figure 1.

The second example output is a grid display of the hydrocarbon emission in which the amount of generated pollutants is displayed numerically in each grid cell. For the purpose, a new graphic parameter file is created with GRPARAM in the following way:

```
Enter: Name of existing parameter-file for default values (optional) =
Enter: Name of parameter-file to be generated = hc.par
Enter: Horizontal[,vertical] grid distances (1,1) = .5 .5
Draw complete grid (n) ? n
Enter: Grid color (4) =
```

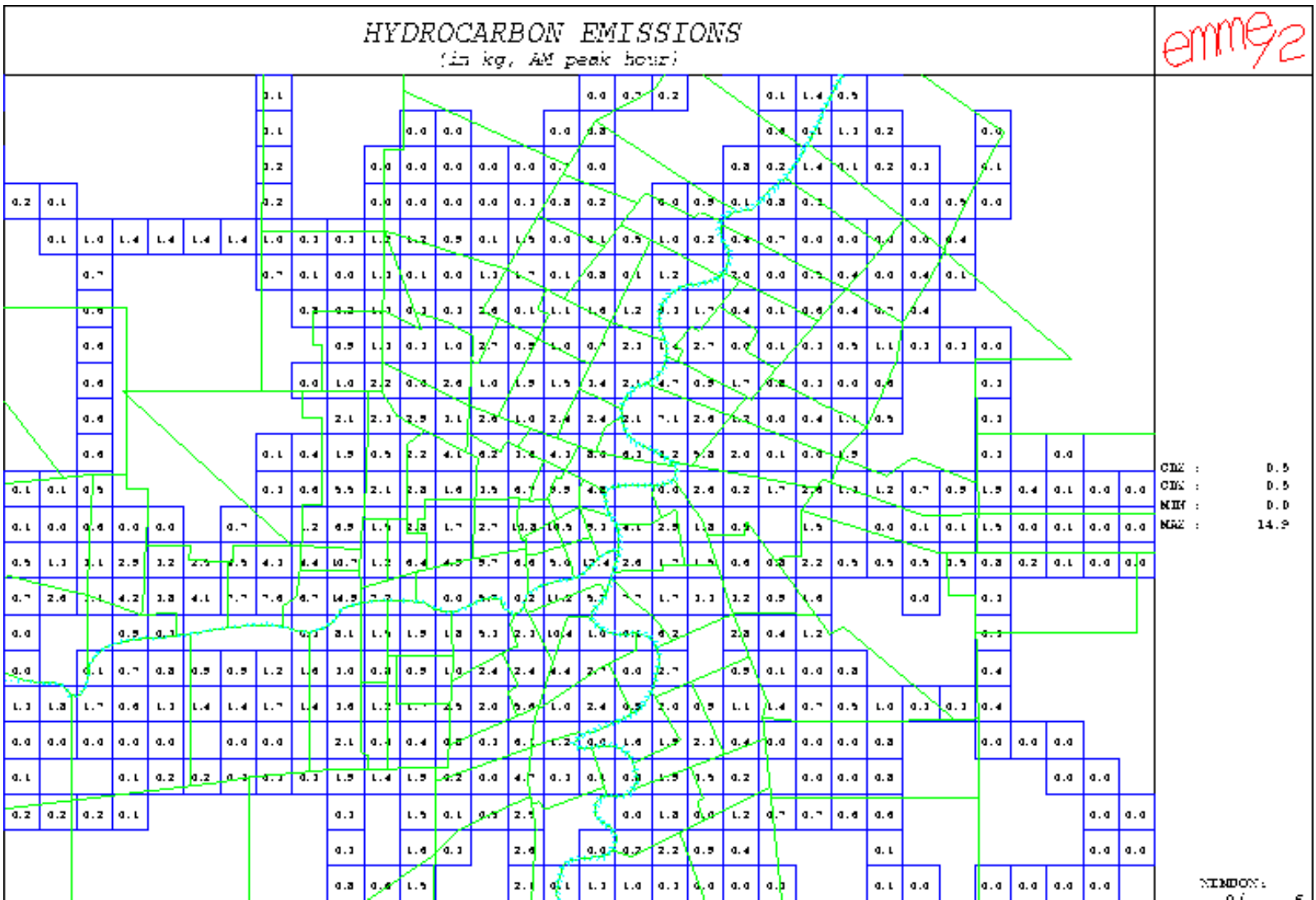
```
Enter: Name or number of data item to be displayed (1) = @hc
Select: Type of gird representation
        1= Classes defined by value interval
        2= Numeric values
        3= Hatching with proportional density
       (1) 2

Enter: Color of the numbers (4) = 2
Enter: Text size of the numbers (*) = -.12
Enter: Minimal field width (*) = 3
Enter: Number of digits after decimal point (*) = 1
Enter: Offset of the number inside grid box (xdisp, ydisp) (*,*) = .26 .17
```

Note that the negative character size implies horizontal centering at the specified offset.

The resulting graphic parameter file `hc.par` contains the following information:

```
GD          .5          .5 Horizontal,vertical grid distance
CG           0             Draw complete grid
GC           4             Grid color
DN        @hc              Name or number of data item to be displayed
RP           1             Representation (0=Class,1=Numbers,2=Proportional)
PC           2             Color of the numbers
NS        -.12             Text size of the numbers
DT           3             Minimal field width
DF           1             Number of digits after decimal point
DG         .26         .17 Offset of the number inside grid box (xdisp, ydisp)
```



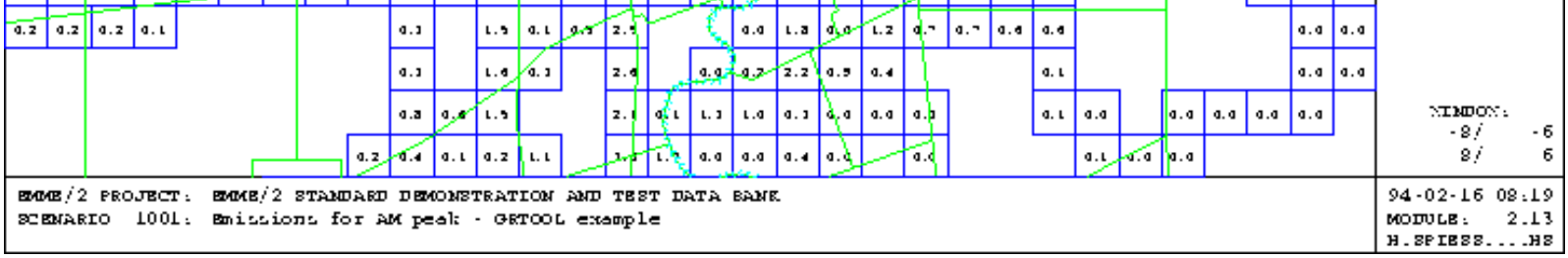


Figure 2: Example of grid representation using numerical values

Running the command

```
grannot hc.par emission.grd annoth
```

produces the annotation `annoth`. This annotation is shown in Figure 2; it is displayed here without the underlying network, but with the zone boundaries instead.

Finally, to finish this series of output examples, a graphic parameter file is created to represent the nitrous oxides levels using proportional hatching. In this type of representation, the number of hatch lines in each grid cell is proportional to the corresponding data value. A scale factor 1 is used here, implying that each vertical line corresponds to 1 kg of pollutant. The following shows the dialog generated by GRPARAM:

```
Enter: Name of existing parameter-file for default values (optional) =
Enter: Name of parameter-file to be generated = nox.par
Enter: Horizontal,vertical grid distance (1,1) = .5 .5
Draw complete grid (n) ? n
Enter: Grid color (4) = 4
Enter: Name or number of data item to be displayed (1) = @nox
Select: Type of gird representation
        1= Classes defined by value interval
        2= Numeric values
        3= Hatching with proportional density
       (1) 3

Enter: Scale factor () = 1
Enter: Color of pattern (4) = 3
```

The resulting parameter file `nox.par` contains the following records:

```
GD          .5          .5 Horizontal,vertical grid distance
CG           0             Draw complete grid
GC           4             Grid color
DN        @nox             Name or number of data item to be displayed
RP           2             Representation (0=Class,1=Numbers,2=Proportional)
SC           1             Scale factor
PC           3             Color of pattern
```

Running the command

```
grannot nox.par emission.grd annotn
```

produces the annotation `annoth`. This annotation is shown in Figure 3; it is displayed here on top of a bandwidth plot of the auto volumes produced with EMME/2 module 6.12.

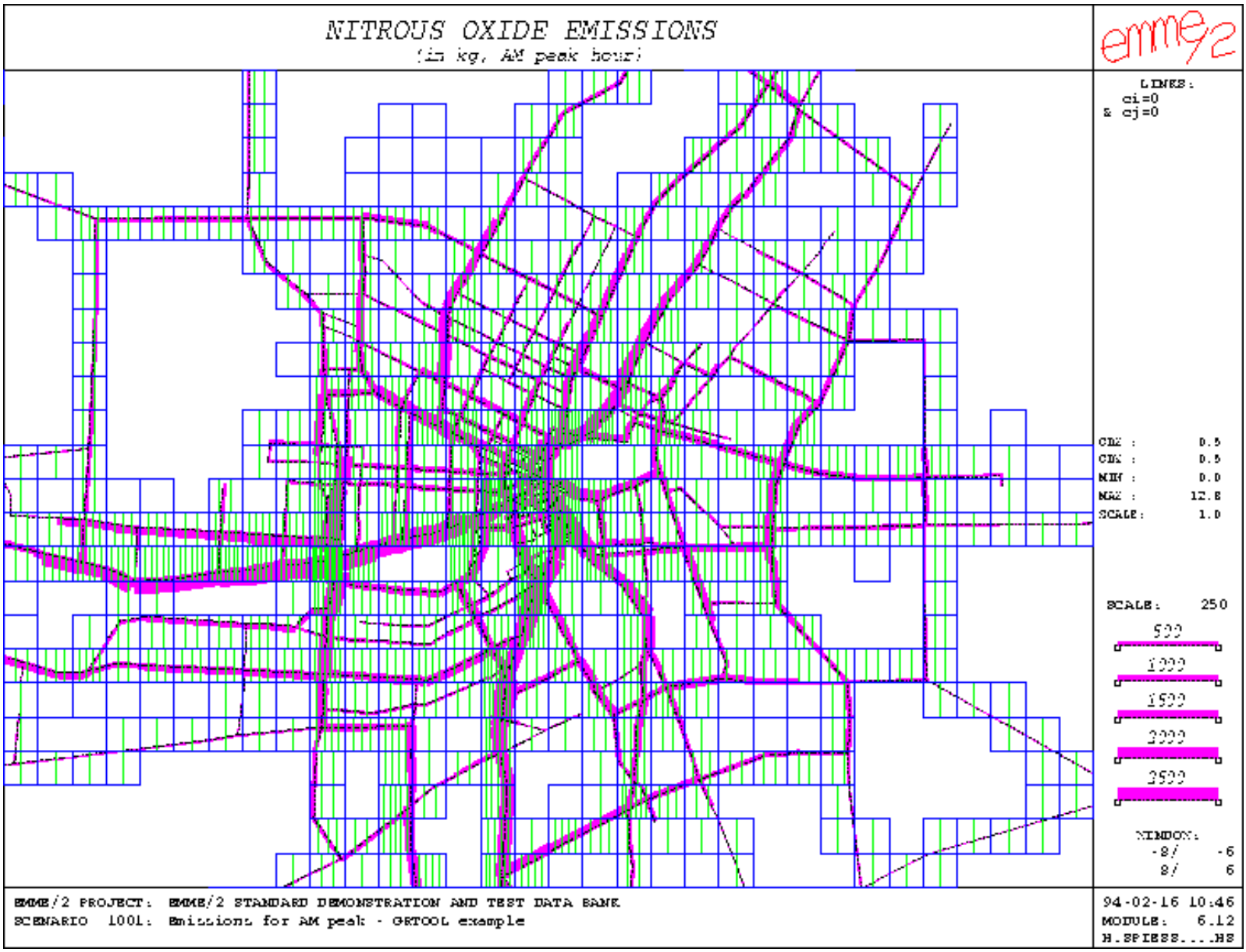


Figure 3: Example of grid representation using proportional hatching

---

Next Up Previous Contents
Up: GRTOOL A Link-to-Grid **Previous:** Annotation File

Heinz Spiess, EMME/2 Support Center
Sun Mar 3 22:21:06 MET 1996