

SCEPTRE

Installation Manual

Wolf-Rainer Novender

Summer 2010

Contents

1	Installation	2
1.1	OS and requirements	2
1.2	Files and directories	2
1.3	Modifications	2
1.4	Make	3
1.5	Un-install	3
2	Use	4
2.1	Overview	4
2.2	SCEPTRE	4
2.3	NGP	5
2.4	Libraries	6
2.4.1	Model library modellib.dat	6
2.4.2	User functions library 'usrlib.a'	7
2.5	CONTINUE run	7
2.6	Other features	7
3	Hints	8
3.1	General remarks	8
3.2	Using models	8
3.3	Error messages from XDIODE	9
3.4	Running NGP	9
3.5	Fortran Compiler — variable names	9
4	Documentation	10
5	Differences	10
6	Contact	10

SCEPTRE (System for Circuit Evaluation and Prediction of Transient Radiation Effects) is a general purpose circuit analysis program, which provides all three major analyses, AC, DC, and transient analysis, on either linear or nonlinear networks. It employs a free-form input language and state variable methods to simulate problems of interest to electrical engineers [1, 2].

Other highlights:

- Flexible nonlinear input. Circuit elements may be described as constants, tabular data, or arbitrary functions of other network quantities.
- Flexible modelling capability. The arbitrary functional capability allows any device to be modelled to the degree of accuracy required for an analysis.
- State-of the art numerical methods. The code simulates networks in a cost-effective manner. It employs three explicit and one implicit integration routines.
- Stored models. In addition to built-in elements, SCEPTRE is capable of storing circuit models permanently, or temporarily in its own model library.
- Automatic initial conditions. The initial conditions of circuit variables are determined in two ways: automatically determined from DC steady-state solution of a circuit network, or manually from user specified initial conditions.
- Rerun. Multiple case rerun based on a single master run may be carried out automatically. The user supplies only the changes that apply from the master run for each repeated run.
- Defined Parameters. A special section has been created to enable the user to define quantities that may be output other than sources or passive currents and voltages. The user may enter systems of first-order differential equations that may or may not have anything to do with a particular electrical network.
- Minimisation of computational delays. The sequencing of instructions by SCEPTRE ensures that computational delays in transient analyses will be few even when complex functional dependencies are present.
- Variety of run controls. The user may control simulations to a fine degree rather than "giving" the network to the code and losing control.
- Subprogram Capability. The user who is familiar with computer programming may write FORTRAN subroutines and insert them in otherwise conventional SCEPTRE runs. This option permits handling special situations, even though these should be rare.
- Multiple analysis available. Allow the analyst to use one code to run various analyses on circuit: DC (sensitivity, worst-case, Monte-Carlo, optimisation), AC, transient analyses.

A graphical user interface NGP has been added using GNUPLOT. As the source code is available, NGP may be modified due to other graphical software.

SCEPTRE and NGP are free software. See the file COPYING for copying permission.

1 Installation

1.1 OS and requirements

- Fortran compiler (f77, g77, gfortran, ...) This version of SCEPTRE has been compiled with gfortran-4.3 (Ubuntu) gcc version 4.3.3 (Ubuntu 4.3.3-5ubuntu4). The Fortran compiler is obligatory to install *and to run* SCEPTRE!
- GNUPLOT ¹ version ≥ 3.7 . You may use any other graphics, but then you have to rewrite the program 'ngp_gnu.f' in the directory 'x3'. You may run SCEPTRE without graphics but the quality of presenting the results is restricted to printer plots.

This version of SCEPTRE has been developed under UNIX/Linux using the SuSE 6.3 distribution. It has been tested (many, many years ago) under different OS like HP-UX, AIX, VMS, MVS, VM, etc. As SCEPTRE has been rewritten completely in ANSI Fortran-77 (more or less) there should be no problems to compile it under different OS. Due to the increasing number of system routines, problems may arise with SCEPTRE subroutines having the same name. The EXTERNAL statement may help (e.g. for subroutine 'x1/getarg.f').

None of the subroutines contain a SAVE statement. Be aware to specify an appropriate compiler option (e.g. '-fno-automatic' or '-static').

System dependent subroutines like date, time, cpu-time, elapsed time, etc. are placed in an extra directory 'x3'.

1.2 Files and directories

Un-tar and uncompress the distribution file in any free directory, e.g. 'sceptre':

```
mkdir sceptre
cd sceptre
tar xvzf sceptre-<version>.tar.gz
```

This will result in directories according to table 1.

1.3 Modifications

Before you install, check the correct call to and options for your Fortran compiler and the final installation directory.

The defaults are:

final installation directory: /usr/local

compiler and options: g77 -fno-automatic -O

If you want to change the final installation directory, modify

sceptre/Makefile: INSTDIR

¹[<http://www.cs.dartmouth.edu/gnuplot_info.html>](http://www.cs.dartmouth.edu/gnuplot_info.html)

Table 1: SCEPTRE directories for the installation

sceptre	top directory (Makefile, COPYING, this file, etc)
sceptre/x1	sources for phase 1 (exec1)
sceptre/x2	sources for phase 2 (libx2.a, exec2.o)
sceptre/x3	sources for auxiliary routines (libx3.a) and NGP
sceptre/bin	shell scripts and executables
sceptre/lib	libraries (usrlib.a, modellib.dat)
sceptre/samples	sample files
sceptre/doc	German and english user manuals (PDF)
sceptre/etc	configuration file for NGP (ngp.rc)

sceptre/bin/sceptre: BASEDIR

sceptre/bin/sceptre.inst: BASEDIR

sceptre/bin/ngp: BASEDIR

If you want to change the compiler, modify

sceptre/Makefile: FC, FFLAGS

sceptre/bin/sceptre: FC

Note: the compiler options in the Makefile are used for the installation only and therefore executed only once. The options in the shell script 'bin/sceptre' are used for every SCEPTRE run.

1.4 Make

All actions are controlled by the Makefile in the top directory. To compile and create all necessary files change to the top directory 'sceptre' and type 'make'. If everything works fine you can install all files with 'make install'. You must be root! Check the installation by running SCEPTRE and NGP on the examples in the directory 'sceptre/samples' (see next paragraphs 'Use'). If the installation meets your requirements you can save disk space with 'make clean'. This will erase all object files and files, which have been copied to the installation directory.

1.5 Un-install

To remove all files from the installation directory type 'make uninstall'. You must be root!

2 Use

2.1 Overview

Each user should have own libraries (usrlib.a, modellib.dat) and a configuration file (ngp.rc), which should reside in \$HOME/lib and \$HOME/etc respectively. For the first use of SCEPTRE/NGP each user should issue the command 'sceptre.inst' which will provide copies of these files as templates in his home directories.

Simulations with SCEPTRE are performed in several steps:

- use any editor and write your SCEPTRE statements in a file, which must have the suffix '<name>.d'.
- execute SCEPTRE by typing 'sceptre <name>'
- follow the interactive instructions (see paragraph SCEPTRE).
- if the simulation was successful and the user has requested a transient or frequency run, SCEPTRE creates a direct access file <name>.dirac, which may be viewed with the graphical postprocessor NGP by typing 'ngp <name>' (see paragraph NGP).

2.2 SCEPTRE

SCEPTRE is started with the command

```
sceptre <name>
```

where 'name' points to an input file '<name>.d', which must contain the SCEPTRE input. After processing the input, the pager 'less' is used to view the output listing '<name>.ls1'. Watch the last line of this listing (using 'G') to verify, that there are no errors. To proceed enter 'Q'. Next the Fortran-Compiler and Linker are invoked. For possible compiler errors see paragraph 'Hints'.

After successfully loading the second phase, EXEC2, the simulation starts. For TR and AC analyses the actual contents of all output variables are shown at the screen unless suppressed by the SCEPTRE statement

```
X PLOT DIMENSION = 0
```

under the RUN CONTROLS heading.

At the end of each simulation the run must be continued with the command 'go' (watch the system messages). Finally the simulation output '<name>.ls2' is listed again with the pager 'less'.

To print the output listings with the correct carriage control use the command

```
asa < name.ls? | lpr -P...
```

where ? must be replaced by '1' or '2' and '-P...' is the printer name for ASCII-files.

See also paragraph 'Hints'.

2.3 NGP

NGP is a very basic program, which processes the output data from a SCEPTRE run, residing on a direct access file, and creates the input data for GNUPLOT.

In case GNUPLOT is not available, any other graphic programs may be used. The source of NGP (ngp_gnu.f) is available and may be changed according to any other graphic software available.

NGP offers all variables, specified in a SCEPTRE run (and reruns) under OUTPUTS, including the independent variable TIME and FREQ. These variables are listed together with an index number (see also at the end of the SCEPTRE listing: 'directory of direct access file').

Several commands are available to draw up to 6 curves in a single plot. For ease of use, variables are referred to by their corresponding index numbers. Any output variable may be specified as independent (X) variable, up to 6 output variables as dependent variables (Y).

Commands (may be specified in upper or lower case):

Select variables: S n0 n1

select output variable with index number 'n0' as X variable, 'n1' etc. as Y variable(s)

Plot/Draw: P or D

creates data files and a GNUPLOT control file, calls GNUPLOT and creates the plot according to the device specification 'term:' in the file \$HOME/etc/ngp.rc

Hardcopy: HC

creates data files and a GNUPLOT control file, calls GNUPLOT and creates the plot according to the device specification 'hc:' in the file \$HOME/etc/ngp.rc

Scaling: SCALE ON|OFF

scales all output quantities to +/- 100 % (or resets it).

List: L

lists all selections

Quit/End: Q or E

terminates NGP

A separate configuration file, 'ngp.rc', which must reside in the directory \$HOME/etc, may be used to configure the output devices for GNUPLOT and the appearance of the plots. It contains 3 entries terminated by ':'. After the terminator any GNUPLOT commands may be entered (refer to the GNUPLOT manual), separated by ';' up to column 72.

A copy of 'ngp.rc' is placed in \$HOME/etc as template, if one calls 'sceptre.inst'.

Example for 'ngp.rc':

```
hc: set term post land; set out "|lpr"
term: set term X11
cmd: set grid; set time
```

In this example the standard plot appears on the screen under X11 when using the P or D command. The HC command converts the plot to PostScript and sends it directly to the PostScript printer (You should press <ENTER> again to get the printer output immediately). Each plot contains a grid and the day and time when it was created.

NGP is invoked by

```
ngp <name>
```

where <name> denotes the name of the SCEPTRE file '<name>.d'. NGP is looking for the direct access file '<name>.dirac', which is only created by a successful SCEPTRE run with TR or AC analysis.

Example for a NGP session:

```
ngp a13
enter command:
s 3 7 8 12
enter command:
p
<ENTER>                                <---- important !!!!
enter command:
hc
<ENTER>                                <---- important (see below)
enter command:
Q
```

This will cause NGP to create a plot with variable '3' as X, and variables '7', '8', and '12' as Y variables. The plot is shown on the device according to the entry under 'term:' in the ngp.rc file. Next a hardcopy is created using the information from the 'hc:' entry.

Note: After the P/D command a new window is opened showing the graphics. To enter new NGP commands, the user must press <ENTER> in the NGP window. In order to continue after a HC command, press <ENTER> in the NGP window.

NGP creates several files with the suffix '.GNU'. The file names consists of the symbolic outputs quantities used in SCEPTRE. Each file consists of tabular data x-variable versus y-variable and may be used also by other graphical programs. The control file for GNUPLOT (the last one executed) is named '<name>.GNU'.

See also paragraph 'Hints'.

2.4 Libraries

2.4.1 Model library modellib.dat

Models may be stored permanently by SCEPTRE using the appropriate statements. For the first model ever stored this library is created automatically in the directory \$HOME/lib, provided that the correct syntax is used

```
MODEL DESCRIPTION (INITIAL)
```


Any existing file `modellib.dat` is erased! This is only true, if the user has called the command `sceptre.inst`, which automatically copies a sample `modellib.dat` as a template into `$HOME/lib`. If no `modellib.dat` exists in `$HOME/lib`, SCEPTRE allocates the model library from the installation directory (e.g. `/usr/local/lib`), which in turn will cause permission problems if the user wants to modify or delete this library.

The created file is readable (ASCII), but should not be edited by any editor. In case of modifications just delete and add the model (see manual and paragraph 'Hints').

2.4.2 User functions library 'usrlib.a'

User may specify their own FORTRAN functions (see manual). These functions must be separately pre-compiled and stored in a library 'usrlib.a', which should be located under `$HOME/lib`. The following steps have to be performed to store 'program.f' with the g77 Fortran compiler:

```
g77 -c -O program.f
ar -rv $HOME/lib/usrlib.a program.o
```

This user library may also be used to replace modified system subroutines like 'post.f' or 'intplt.f'. To incorporate any system modifications it is important, that 'usrlib.a' is searched before 'libx2.a' in the link step (see shell script 'bin/sceptre'). An empty 'usrlib.a' is copied to `$HOME/lib` when calling 'sceptre.inst'. If this library is missing, SCEPTRE allocates it from the installation directory.

2.5 CONTINUE run

In order to use the CONTINUE feature (see SCEPTRE manuals), the user has to create a new SCEPTRE input file, which must contain at least the following statements:

```
CONTINUE
RUN CONTROLS
    STOP TIME = <new stop time>
END
```

where <new stop time> denotes a new value for the STOP TIME, which must be greater than the STOP TIME of the run, which is to be continued. SCEPTRE is started in the usual way with the name of the input file (without suffix '.d') as parameter. It then looks for the files 'pgmsaves.for' and 'datasave.dat' which should exist from the preceeding run. This makes clear that the CONTINUE run must immediately follow the master run (or the files 'pgmsave.for' and 'datasave.dat' must be saved).

2.6 Other features

There are two features described in the manuals, which have not been tested:

- REOUTPUT (to my opinion this feature does not make sense any more)
- convolution mode.

3 Hints

3.1 General remarks

- All SCEPTRE keywords must be entered in CAPITAL LETTERS. Do not use any special characters like TAB, ESC, etc. within the input file. The input may not exceed column 72.
- All text after a semicolon ';' will be treated as comment, even in SCEPTRE comments (after MODEL/CIRCUIT DESCRIPTION).
- The printer plots are still available. They may be invoked by the keyword PLOT (under OUTPUTS). For composite plots PLOT INTERVAL must be specified under RUN CONTROLS. See manual for detailed information.
- Independent from these options all output variables are stored onto a direct access file, which may be viewed and processed with the stand-alone post processor NGP (see separate description).
- To speed up the simulation considerably specify X PLOT DIMENSION = 0 under RUN CONTROLS. This will suppress the simultaneous output on the screen during simulation.
- Please study the hints in appendix A (english manual) and in chapter 'Allgemeine Hinweise' of the german manual carefully.

3.2 Using models

Some of the examples in the directory `sceptre/samples` make use of the permanent model library 'modellib.dat'. A copy of this library will be placed in the directory `$HOME/lib` when 'sceptre.inst' is called.

If you want to setup a new permanent library with new models, you have to use the entry

```
MODEL DESCRIPTION (INITIAL)
```

only for the *very first* run. This will create a new file 'modellib.dat', regardless if it already exists or not!! If you want to modify an existing library 'modellib.dat', you must not specify (INITIAL). The correct entry for this case would be just

```
MODEL DESCRIPTION
```

Remember: you may add and delete models in the permanent library very easy (see manual).

You may specify output requests under OUTPUTS in models. These outputs are *added* to the output specification under CIRCUIT DESCRIPTION. Usually you never enter OUTPUTS under the MODEL DESCRIPTION.

When using models ensure that model and main circuit use the same set of units. It is a good practice to enter the chosen units as a comment in both, MODEL and CIRCUIT DESCRIPTION.

3.3 Error messages from XDIODE

Using DIODE EQUATION with the integration routine IMPLICIT may cause numerical overflow in the subroutine XDIODE. In the original IBM version this overflow was handled by the subroutine CALL ERRSET(262,300,10,2), which suppresses the messages and replaces the result with the highest floating number. The overflow occurs when SCEPTRE evaluates the diode current for the actual integration step size as a first approach. Due to the overflow the step size is reduced until the result conforms to the given error criteria. This 'error' has no influence to the accuracy of the results.

3.4 Running NGP

NGP is a very basic program to extract data from the direct access file and feed them to GNUPLOT. NGP takes the first comment line after CIRCUIT DESCRIPTION and inserts it as title with the GNUPLOT command 'set title "....."'. Therefore the user should enter at least one comment line (up to 11 are allowed) after CIRCUIT DESCRIPTION, but this line must not contain any double quotes. Otherwise the parser of GNUPLOT will get confused.

Remember: Neither printed nor graphical outputs contain any physical units! SCEPTRE does not know, what set of units has been chosen (see manuals).

3.5 Fortran Compiler — variable names

Unfortunately user chosen variable names may clash with internal program variable names. This is detected by the Fortran compiler. Avoid the defined parameter name

- PS (and DPS respectively).

Same applies to the dummy arguments in an EQUATION statement, e.g.

```
Q... (A, B, C, ...) = .....
```

which will be converted to a Fortran statement function with the same dummy variable names A,B,C,... Some Fortran compiler get confused if the program uses internal variables with the same name. Under certain circumstances SCEPTRE generates internal array names A(...), B(...), etc. which will result in a compile time error. To avoid this, use EXPRESSION statements instead or define the dummy variables in the equation as

```
Q... (AA, BB, CC, ...) = (...AA...BB...CC...)
```

To make your SCEPTRE input independent from any Fortran Compiler do not distinguish variable names by upper and lower case characters. SCEPTRE will distinguish between the variables, e.g. 'PMI' and 'Pmi'. But depending on the Fortran compiler all variable names may be converted to upper cases causing run time errors due to doubly defined variables.

4 Documentation

There are two user manuals as PDF in the directory `sceptre/doc`:

1. `doc_en.pdf`: a copy of the original 'Extended SCEPTRE User's Manual', Vol.1 from D. Becker with some minor revisions.
2. `doc_ger.pdf`: a German manual, which I wrote and which is based upon my long experience with SCEPTRE. It is not a translation of the english manual, but it contains almost the same topics and examples as the original manual. Additionally it describes the graphical interface routines 'intplt.f' and 'post.f' for the postprocessor NGP as well as the periodic functions FPULSE and FSINSQ.

5 Differences

This UNIX/Linux version has been completely rewritten from the original SCEPTRE IBM version 9/73, S/360 release 21.7.

The UNIX/Linux version differs from the original version in a few topics:

- SCEPTRE ignores any information after a semicolon ";". This feature may be used to add comments to SCEPTRE input lines (does not work well under RERUN DESCRIPTION);
- the key word PRINT in model calls prints only the included model instead of the whole library;
- the MAXIMUM PRINT POINTS have been defaulted to zero;
- instead the values of all output quantities are printed at the START TIME and STOP TIME ('initial and final values of output variables');
- the number of defined parameters and their derivatives (P..., DP...) has been increased from 100 to 200;
- statistics about all output quantities (maximum, minimum, solution points) and directory of the direct access file.

6 Contact

For additional information please contact me via e-mail or snail-mail:

Prof. Dr. Wolf-Rainer Novender
FH Giessen/Friedberg
Fachbereich IEM
Wilhelm-Leuschner-Str. 13
D-61169 Friedberg
Germany

e-mail: `wolf-rainer.novender@iem.fh-friedberg.de`

References

- [1] J.C. Bowers, S.R. Sedore. *SCEPTRE: A Computer Program for Circuit and System Analysis*. Englewood Cliffs, N.J., Prentice-Hall, Inc., 1971
- [2] R. W. Jensen, L. P. McNamee. *Handbook of Circuit Analysis Languages and Techniques*. Englewood Cliffs, N.J., Prentice-Hall, Inc., 1976