# SONATA – A SYSTEM OF NEARRINGS AND THEIR APPLICATIONS

THE SONATA TEAM

ABSTRACT. SONATA (System of Nearrings and their Applications) is an implementation of methods for the construction and analysis of nearrings into the group-theory system GAP. It consists of three main parts:

1. A library of all small nearrings (up to order 15). Along with the library come various functions for the analysis of the structure and the elements of these nearrings.
2. Fast algorithms for constructing the following nearrings.
   (a) nearrings of polynomial functions on groups,
   (b) d. g. nearrings as e. g. $I(G)$, $A(G)$, $E(G)$,
   (c) centralizer nearrings
3. Planar nearrings and BIB-designs from those.

## 1. MOTIVATION

When working on our master's and PhD-projects in near-ring theory we hoped that we would gain more insight into the problems we worked on if we had significant examples at hand. For example, Erhard Aichinger wanted to find 1-affine complete groups; these are groups on which every unary compatible, i.e., congruence preserving, function can be interpolated by a polynomial function. In other words, a group $G$ is 1-affine complete if the near-ring $I(G)$ contains all functions in the near-ring $C_0(G)$ of all zero-preserving compatible functions. After having written a straight-forward program to compute all polynomial functions on a group, which basically relied on computing all terms and checking whether the arising functions were equal, he found that in that way probably only the groups of order less than 10 could be treated, and therefore abandoned his hope to find help in computers for quite a while.

At about the same time, Christof Nöbauer was collecting a library of *all* small near-rings; and he decided to implement his library into the group-theory system GAP. Then Jürgen Ecker started to break rings into their subdirectly irreducible parts using GAP.

In May 1995, we realized that the problem of computing the number of polynomial functions on a group was actually an easy task if one used the power of computational group theory. The easy key observation is that it is easy to compute how big the group generated by some group elements is. Representing the functions on $G$ as elements of $G^{|G|}$, it is easy to compute first the generators of the group of polynomial functions and then all polynomial functions as the closure of these generators. This observation, albeit strikingly easy, and of course not even original, made it possible to compute the number of polynomial functions on $S_4$, which is 22

265 110 462 464, in a few seconds. The same strategy also worked for other kinds of distributively generated near-rings, such as $A(G)$ or $E(G)$.

At this point, we decided to make a package of our functions and make them available to a wider community. Encouraged by the enthusiasm of Prof. Günter Pilz, and paid by the "Fonds zur Förderung wissenschaftlicher Forschung", we started to bring our functions into a common form, and to add many functions that we found useful, as e.g. the computation of Noetherian Quotients, which works especially fine for polynomial near-rings. We wanted to include the applications of near-ring theory to design-theory, because especially in this field we thought that it could only be through the examination of examples that the contribution of near-rings to this field could be investigated. This part was then mainly carried forward by Roland Eggetsberger and Peter Mayr.

In the beginning of 1997, with the conference at Stellenbosch coming near, we decided to make our programs available to near-ringers six months later. Despite of the fact that SONATA does not contain many new sophisticated algorithms, and therefore hardly represents a big deal in computational near-ring theory, it uses a lot of well-established sophisticated algorithms in group theory. We think that our real contribution is to take advantage of these algorithms for computing with near-rings. Nevertheless, computational near-ring theory could be interesting: A typical problem arising in the computation of near-rings would be the following: Given a function on a group, how big is the near-ring generated by it? And what if we take more than one function? And how can a given function in the near-ring be represented by the generators. We recall that Sim's stabilizer chains give solutions to similar problems in the theory of permutation groups. We think that even an answer for one function, and on special groups, would be delightful.

Examples of near-rings are nice; but it would be even nicer to have a lot of interesting examples in a small booklet. At this point, Franz Binder joined us, and started to work on a near-ring table containing all near-rings up to a certain order and giving meaningful information about each of them, such as for example the ideal lattice with commutators in the sense of universal algebra. It was when he started to work on this complete library that the people at the Maths Department, whose printers we constantly fed with new near-ring information, started to give us rather strange looks, which we could not explain to ourselves but as signs of starting admiration for the beauty of near-rings.

When a beta version of GAP4 came out, we found that SONATA should be written in this new version of GAP4. Jürgen Ecker's effort to translate all existing GAP3-code into GAP4 was rewarded by the observation that many things worked much better in GAP4 than they did before. Just before leaving to Stellenbosch, he invented the name SONATA for our programs that sometimes proceed *andante*, but at other times really rather *presto*.

At the near-ring conference in Stellenbosch in July 1997, we showed some possibilities of our system to many people doing research in near-ring theory. Their interest in our system showed us that our hope that near-ringers would actually use SONATA was by no means unsubstantiated.

Finally, with October 1st, 1997, we give away the first version of SONATA. We are eager to hear YOUR feedback in order to make SONATA nicer in all respects.

We, the SONATA team, want to say THANK YOU to all that have helped us in some way to realize this project:

1. to Tim Boykett for helping us with his expertise in computers and how they could be used in algebra;
2. to Marcel Widi, who contributed some functions on semigroups which, however, do not lie in the scope of the SONATA project.
3. to Christof Nöbauer, who not only filled our hard disks with an evergrowing number of near-rings, but also worked a lot to keep our computers alive.
4. to the staff at the algebra group at our department, and in particular to all our visitors, who have put lots of ideas into our heads: this includes Peter Fuchs, Gerhard Betsch, Jim Clay, Carl Maxson, Gary Birkenmeier.
5. to the Forschungsfonds for financing this project.

Needless to say, this project would never have been carried out without the encouragement and suggestions of Prof. Günter Pilz.

## 2. Where to get GAP 4 and SONATA

2.1. **GAP 4.** First of all you will have to install GAP, Version 4, on your computer. You can get an actual version of GAP 4 via anonymous ftp from the following servers:

- On `ftp.math.rwth-aachen.de` in the directory `pub/gap4b1/`.
- On `ftp-gap.dcs.st-and.ac.uk` in the directory `pub/gap/gap4b1/`.

In order to install GAP 4 Beta 1 under UNIX (a version for DOS will be available in a few weeks) you will need the following files. The size of the unpacked distribution is about 45 MByte.

**INSTALL:** (12 KByte): Installation guide for GAP 4 Beta 1 under UNIX
**gap4b1.tar.gz:** (7.8 MByte): The source and library code for GAP 4 Beta 1.
**doc4b1.tar.gz:** : The documentation including DVI files for GAP 4 Beta 1

Next you have to get all bugfixes:

**fix1.tar.gz:** : This fix corrects several bugs and problems. Some of these errors may cause wrong results, some may cause GAP to issue an error message. See the included file 'description1' for details. You have to apply this bugfix if you want to run SONATA without problems.
**...:** : From time to time there are new bugfixes. New fixes will concern the functions for groups and will not have a very big effect on the functions for near-rings.

2.2. **SONATA.** You can get an actual version of SONATA via anonymous ftp from our server:

- `bruckner.stoch.uni-linz.ac.at` in the directory `pub/sonata/`.

In order to install SONATA you will need the following files. The unpacked distribution will require about 20 MB on your hard disk.

**install.txt:** (1 KB): Installation guide for SONATA.
**sonata.tar.gz:** (710 KB): the library code for SONATA.
**sondoc.gz:** (50 KB): The documentation including a tutorial for SONATA.

## 3. How to install GAP 4 and SONATA

First you will have to install GAP. The file `INSTALL` contains information how to install the program. If you have succeeded in installing GAP, the trickiest part is done (and you only have to do it once).
The installation of SONATA requires not more than 4 or 5 easy UNIX-commands.

## 4. How to start GAP and SONATA

Once GAP and SONATA are installed you can start GAP with

```
gap4
```

and at the GAP prompt

```
gap>
```

type

```
gap> RequirePackage( "sonata" );
```

Now all the functions for near-rings are available for you in GAP. For your first steps with GAP and SONATA you may consult the SONATA-Tutorial.

## 5. How to learn GAP 4 and SONATA

The first steps are usually the most complicated. The easiest way is to print out the SONATA-tutorial "9 easy pieces for SONATA". You can read the tutorial on the next few pages. A few easy examples show, how GAP and SONATA can be used to answer some nontrivial questions. You can use them as a basis for your own attempts. As you are getting more familiar with the system, feel encouraged to study the manuals of SONATA and GAP and find out about the more involved features. Note: there is definitely nobody in the world who knows every function of GAP and SONATA and you will never need to know all functions at a time.

## 6. Troubles?

SONATA is not a commercial program and we do not guarantee that the results computed with the help of SONATA or GAP are always accurate. If you find errors we would like to be informed so that we can improve SONATA. Please send us the lines that produced the error together with the numbers of your GAP and SONATA versions. If you encounter any troubles when installing or running SONATA, do not hesitate to ask us. Our e-mail adress is

```
sonata-trouble@bruckner.stoch.uni-linz.ac.at
```

So, what you have now is a system that conatains a library of all small near-rings and many functions to construct and analyze a lot of interesting big near-rings. Have fun !

the SONATA Team