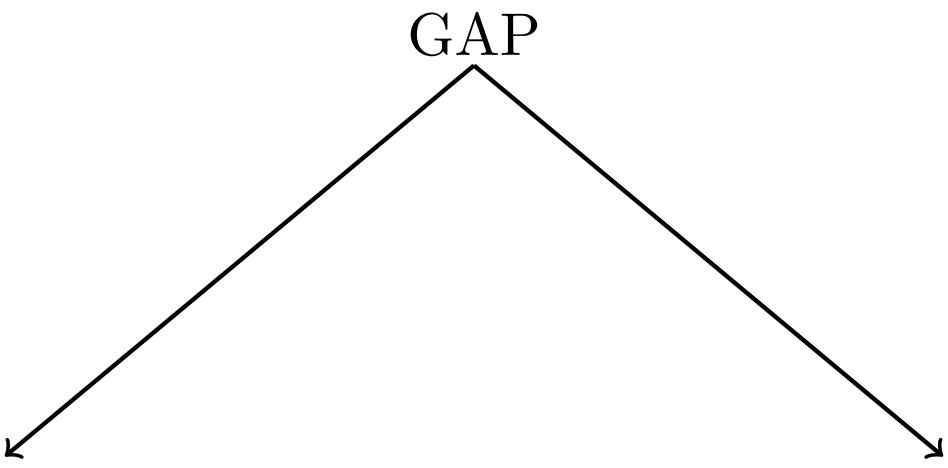
GAP Packages and Libraries: Using existing GAP infrastructure

GAPDays Summer 2025

Meike Weiss and Lukas Schnelle

August 2025

Overview of GAP



Core GAP

e.g. groups, lists,...

Packages

e.g. graphs, projective spaces, crystallographic groups,...

Packages

Why are they helpful?

Packages

Why are they helpful?

1. Additional functions and objects for specialized purpose
2. Libraries of different objects

Packages

Why are they helpful?

1. Additional functions and objects for specialized purpose
2. Libraries of different objects

Where can you find the packages?

- Shipped Packages:
  `https://www.gap-system.org/packages`

- More packages: `https://github.com/gap-packages`
- Even more packages exist

Loading Packages

- In GAP session:

```
gap> LoadPackage("Grape");
-------------------------------------------------------------------------------
Loading GRAPE 4.9.2 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (https://webspace.maths.qmul.ac.uk/l.h.soicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
true
```

Loading Packages

- In GAP session:

```
gap> LoadPackage("Grape");
-------------------------------------------------------------------------------
Loading GRAPE 4.9.2 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (https://webspace.maths.qmul.ac.uk/l.h.soicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
true
```

- Automatic loading is possible defined e.g. in *gap.ini*:

```
 GAP   GAP 4.14.0 of 2024-12-05
       https://www.gap-system.org
       Architecture: x86_64-pc-linux-gnu-default64-kv9
Configuration:  gmp 6.2.0, GASMAN, readline
Loading the library and packages ...
Packages:   AtlasRep 2.1.9, AutoDoc 2023.06.19, Browse 1.8.21, CTblLib 1.3.9, datastructures 0.3.1,
            Digraphs 1.10.0, FactInt 1.6.3, FGA 1.5.0, Forms 1.2.12, GAPDoc 1.6.7, genss 1.6.9,
            GRAPE 4.9.2, IO 4.9.1, NautyTracesInterface 0.3, orb 4.9.1, PrimGrp 3.4.4, recog 1.4.3,
            SmallGrp 1.5.4, SpinSym 1.5.2, StandardFF 1.0, TomLib 1.2.11, TransGrp 3.6.5,
            utils 0.85
Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap>
```

Loading Packages

- In GAP session:

```
gap> LoadPackage("Grape");
-------------------------------------------------------------------------------
Loading GRAPE 4.9.2 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (https://webspace.maths.qmul.ac.uk/l.h.soicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
true
```

- Automatic loading is possible defined e.g. in *gap.ini*:

```
 ┌─────┐   GAP 4.14.0 of 2024-12-05
 │ GAP │   https://www.gap-system.org
 └─────┘   Architecture: x86_64-pc-linux-gnu-default64-kv9
Configuration:  gmp 6.2.0, GASMAN, readline
Loading the library and packages ...
Packages:   AtlasRep 2.1.9, AutoDoc 2023.06.19, Browse 1.8.21, CTblLib 1.3.9, datastructures 0.3.1,
            Digraphs 1.10.0, FactInt 1.6.3, FGA 1.5.0, Forms 1.2.12, GAPDoc 1.6.7, genss 1.6.9,
            GRAPE 4.9.2, IO 4.9.1, NautyTracesInterface 0.3, orb 4.9.1, PrimGrp 3.4.4, recog 1.4.3,
            SmallGrp 1.5.4, SpinSym 1.5.2, StandardFF 1.0, TomLib 1.2.11, TransGrp 3.6.5,
            utils 0.85
Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap>
```

→ The packages must already be installed

Package Manager

Goal:

- Easy installation of packages
  $\rightarrow$ Installs also the dependencies
- Easy way to get a newer version of a package

Package Manager

Goal:

- Easy installation of packages
  $\rightarrow$ Installs also the dependencies
- Easy way to get a newer version of a package

Requirement: PackageManager must be loaded

Package Manager

Goal:

- Easy installation of packages
  → Installs also the dependencies
- Easy way to get a newer version of a package

Requirement: PackageManager must be loaded

Example:

```
gap> InstallPackage("Grape");
gap> InstallPackage("https://github.com/gap-packages/grape.git");
gap> LoadPackage("Grape");
```

Documentation

- Each package has a manual (pdf and html version)
- Can be found here:
  `https://www.gap-system.org/packages`

- Provide explanation of the capabilities and examples
- Can be also used by `?` in a GAP session

Technical Remarks

- Packages are stored in the pkg directory of your GAP directory
- Source code is written in .gi files
- Documentation is XML-based and can be used with GAPDoc (also a package)
- Should have tests
- If you find any bugs: Go to the git repository of the package and write an issue

Example

Goal: Computations with simplicial complexes

Steps:

Example

Goal: Computations with simplicial complexes

Steps:

1. Search the GAP packages list for *simplicial complexes*

Example

Goal: Computations with simplicial complexes

Steps:

1. Search the GAP packages list for *simplicial complexes*
   $\rightarrow$ simpcomp

Example

Goal: Computations with simplicial complexes

Steps:

1. Search the GAP packages list for *simplicial complexes*
   $\rightarrow$ simpcomp
2. Open the documentation to see if the package can do what you want to do

Example

Goal: Computations with simplicial complexes

Steps:

1. Search the GAP packages list for *simplicial complexes*
   → simpcomp
2. Open the documentation to see if the package can do what you want to do
   → Yes!

Example

Goal: Computations with simplicial complexes

Steps:

1. Search the GAP packages list for *simplicial complexes*
   → simpcomp
2. Open the documentation to see if the package can do what you want to do
   → Yes!
3. Load the package

Example

Goal: Computations with simplicial complexes

Steps:

1. Search the GAP packages list for *simplicial complexes*
   → simpcomp
2. Open the documentation to see if the package can do what you want to do
   → Yes!
3. Load the package
4. Try out what is possible

Exercise

Choose a package that you would like to try and do the provided exercise or search for a different package you are interested in:

1. Grape (Graphs and Groups)
2. SimplicialSurfaces (Triangulated Surfaces)
3. FinInG (Finite Incidence Geometry)
4. Automata
5. Guava (Codes)
6. Digraphs (Graphs)

Feedback

What was your experience with the packages?

Did you have any problems?

Do you have any questions for the authors?

Do you miss packages on a specific topic?

Further questions?