Maple Lecture 1. Introduction to Computer Algebra

In this first lecture we define computer algebra and sketch the organization of the course. This corresponds to the first chapter of [6]. The most important aspect of this lecture is the design and organization of worksheets.

1.1 What is Computer Algebra?

Computer Algebra is the discipline that studies the algorithms for Symbolic Computation. In Symbolic Computation, one computes with symbols, rather than with numbers. In this course we are mostly concerned with the practical aspects of Symbolic Computation, in particular its implementation and its application to solve practical mathematical problems. The lecture notes on computer algebra are organized in four parts:

First steps with Maple: We start out using Maple as a calculator, where numbers are not limited to the hardware integers or floating-point numbers, and can be made up of symbols, which is useful to calculate in field extensions.

Polynomials and rational expressions: Computer algebra packages are essentially "equation crunchers" (in analogy with, or contrast to, the number crunching algorithms of numerical analysis). Therefore it is worthwhile to look into the internal data representations and basic expression manipulation facilities.

Calculus: Integration is one of the highlights of computer algebra. The "killer application" which has made Maple into a commercial success is indeed calculus teaching. But we will do more than automating calculus. In Maple we can write procedures which take procedures as input and return procedures.

Advanced Maple: Maple has many composite data types. Besides plotting, we can solve polynomial, differential and linear equations with Maple.

Maple is available as a symbolic computing toolbox in MATLAB. MATLAB is another software system for scientific computation, originally designed for – and still primarily used for – numerical calculations. The fifth part of the course is an introduction to MATLAB ([7] is an excellent guide).

1.2 Computer Algebra Systems

We divide computer algebra systems into general and special purpose systems. Maple [9] is a general purpose system, designed to handle a wide variety of problems. Another general purpose computer algebra system is Mathematica [11]. Whereas Maple and Mathematica are commercial, Axiom [8] is free software. Special purpose systems are created to solve problems in a specific branch of mathematics, physics or engineering. For example, CoCoA [1], SINGULAR [5] and Macaulay [4] are three special purpose systems in the field of computational algebraic geometry. See [3] and [10] for an overview of other systems.

Computer algebra systems are interactive and emphasize on exactness in their calculations, or when that cannot be achieved, on high precision calculations. Other properties are the facilities to manipulate expressions, the wealth of algorithms available, and the plotting capabilities. For instance,

```
[> plot(exp(-x^2)*sin(Pi*x^3),x=-2..2);
```

Imagine the struggle for plotting this by hand.

The goal of Computer Algebra is to automate tedious calculations (but not to replace reasoning by calculation). Maple gives you access to advanced mathematical tools. Rather than struggling to implement the mathematical techniques in some lower level language, Maple offers a nice interface to its kernel and library. In this lecture we learn how to create Maple worksheets. With some effort and discipline, these worksheets can be used very efficiently and effectively to do mathematics.

Sometimes you have to assist Maple to arrive at "good" results. Another problem is expression swell. We give an example, testing the fundamental theorem of calculus: integration is antidifferentiation.

Let us type in the following commands:

```
[> formula := (x^2+1)^25;
[> integrand := diff(formula,x);
[> integral := Int(integrand,x);
[> answer := value(integral);
```

The answer is not quite as expected. Observe that there is no constant term in the expression above. The problem is in the choice of the integration constant.

```
[> factor(answer+1);
```

The example above is also an illustration of expression swell in the intermediate answers of a calculation. The result of the integration can be represented by a compact formula, with a right choice of the integration constant.

1.3 Using Maple in Worksheet and Command Line Mode

A first version of these lecture notes was written using Maple 8. The current version of Maple is Maple 10. Whenever necessary, we will indicate the differences in the releases.

In worksheet mode, we have the choice – at least on PCs running Windows – between the Classic Worksheet (orange icon) and the newer Standard Worksheet (red icon). The Standard Worksheet is a more advanced graphical user interface, you will notice the Palettes, but requires more memory. Compared to the Classic Worksheet, launching Maple in Standard Worksheet mode is noticeably slower on most current computers. For this course, we will almost always use Maple with Classic Worksheets.

A new feature of Maple 10 is the Document Mode. This mode gives a Maple worksheet the same look as any mathematical text on paper, but still with access to the full mathematical engine, so that the document is still very much alive like a computer program. At any time, the user can switch between the Worksheet and the Document mode.

At the opposite end of the possible ways to use Maple is the command line mode. After opening a `Command Prompt` window, one can launch Maple by typing `cmaple10`. Note that, for this to work, it may be necessary to adjust the `PATH` variable in the environment. This command line mode could be very useful when running Maple on a remote server, say via a slow telephone line, when the graphical user interface would become too expensive to use. Another application of the command line mode is the running of batch jobs which do not require any user interaction.

1.4 Design of Maple and Worksheets

Maple consists of four parts: kernel, interface, library, and share library. The share library collects user contributions, available online at the Maple Application Center, visit `http://www.mapleapps.com`. While for efficiency reasons most kernel routines are written in C, the procedures in the library are written in the language of Maple and can be viewed:

```
[> interface(verboseproc=3);      # to show Maple code
[> print(factor);
```

One of the strongest features of Maple is the worksheet. A worksheet can be viewed as a regular text document, but enhanced with computational features like a spreadsheet, or with layers and links like hypertext.

Organization: We organize a worksheet in sections and subsections as we would structure a technical report. With the **insert** menu we place headings for sections and subsections.

Text and Commands: When Maple waits for a command, we see the prompt:

```
[>                              # waiting for a command
```

If instead of a command, we want to insert text, we can remove the `>` with the backspace to obtain

```
[                               # waiting for text
```

We can format texts via the options of the **format** menu.

Execution Groups: Like a paragraph of text is a collection of sentences to be read and understood in the same context, an execution group in Maple is a sequence of instructions to be executed as one. With the `Split or Join` option in the **edit** menu we can respectively break or make execution groups.

Order of Execution: When we load an existing Maple worksheet into memory (see the options in the **file** menu), the worksheet is considered as plain text. We must execute the instructions (via `Execute` from the **edit** menu) to give values to the variables.

The worksheets we make will follow the rules of good worksheet hygiene [2, pages 26-27]. Following is a brief adaptation of these rules:

1. Always begin a worksheet with a `restart` command. When executing an entire worksheet, the `restart` will ensure that all variables are freed from any previous values and settings.

2. While commands in a worksheet may be executed in any order, we will always use a strict linear "top-down" order of the commands in the worksheet. The execution of an entire worksheet is then similar to running a program.

3. Think of a worksheet as a research paper and use titles, sections, and subsections to give a meaningful structure to the order of computations. A typical structure could for example be the derivation of the equations governing a mathematical model, an appropriate choice of algorithm(s), results of running the algorithm(s) on various data, plots visualizing the results, ending with conclusions.

4. Use text to explain what the assumptions and goals are of your computations.

5. Instead of loading an entire package into the worksheet, use the long form, e.g.: `plots[display](...)` when only the `display` command from the `plots` package is needed.

One of the main goals in this course is to learn how to exploit the features of Maple worksheet to design scientific programs at a level as close as possible to mathematical modeling.

1.5 Assignments

1. Follow the New User's Tour (accessible as the third item from the **help** menu).

   Copy to a separate *classic(!)* Maple worksheet those instructions you need to plot the following:
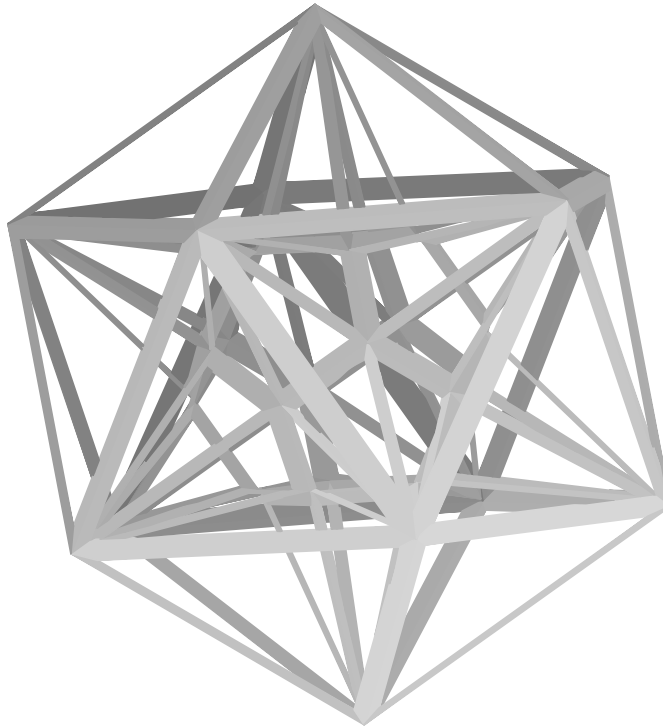
2. How can you see in Maple whether a function is part of the kernel, or written in the language of Maple? Hint: try viewing the code for the Maple commands **diff** and **int**.

3. Explain the difference between **int** and **Int**. Illustrate with an example.

4. Visit `http://www.mapleapps.com` and download one worksheet. Execute it and report on your experiences, answering the following questions: why did you choose this worksheet and what did you learn after executing the worksheet?

5. Make a worksheet with the content of this note. The worksheet must contain all section headings and all the Maple instructions in this note.

6. The command `kernelopts(bytesalloc)` returns the total number of bytes allocated by the Maple mathematical kernel. Execute this command in a Standard Worksheet, in a Classic Worksheet, and in the command line version of Maple. Compare the three values returned by this command and use these to explain why launching Maple in the Standard Worksheet mode is much slower.

References

[1] CoCoATeam. CoCoA: a system for doing Computations in Commutative Algebra. Available at `http://cocoa.dima.unige.it`.

[2] R.M. Corless. *Essential Maple 7. An introduction for Scientific Programmers.* Springer-Verlag, 2002.

[3] J. Grabmeier, E. Kaltofen, and V. Weispfennig, editors. *Computer Algebra Handbook. Foundations, Applications, Systems.* Springer-Verlag, 2003.

[4] Daniel R. Grayson and Michael E. Stillman. Macaulay 2, a software system for research in algebraic geometry. Available at http://www.math.uiuc.edu/Macaulay2/.

[5] G.-M. Greuel, G. Pfister, and H. Schönemann. Singular 2.0. A Computer Algebra System for Polynomial Computations, Centre for Computer Algebra, University of Kaiserslautern, 2001. `http://www.singular.uni-kl.de`.

[6] A. Heck. *Introduction to Maple.* Springer-Verlag, third edition, 2003.

[7] D.J. Higham and N.J. Higham. *MATLAB Guide.* SIAM, 2000.

[8] R.D. Jenks and R.S. Sutor. *AXIOM. The Scientific Computation Systems.* Springer–Verlag, 1992. Download a free version of AXIOM via http://www.nongnu.org/axiom.

[9] Maplesoft. *Maple 10. Getting Started Guide.* Maplesoft, 2005. Available for download via http://www.maplesoft.com.

[10] M.J. Wester, editor. *Computer algebra systems: a practical guide.* Wiley, 1999.

[11] S. Wolfram. *The Mathematica Book.* Wolfram Media, Cambridge University Press, fifth edition, 2003. Online version via http://www.wolfram.com.