

Maple Lecture 4. Algebraic and Complex Numbers

The number $\sqrt[3]{5}$ is often called “irrational” because we cannot represent it as a rational number. Nevertheless, we know $\sqrt[3]{5}$ as a solution to the equation $x^3 - 5 = 0$. Using Newton’s method on this equation we may compute as many decimal places in the approximation of $\sqrt[3]{5}$ as we like. Numbers like $\sqrt[3]{5}$ which are roots of algebraic equations are called *algebraic*. The material of this lecture corresponds to [3, Section 2.5 and 2.6].

The polynomial equation $x^2 + 1 = 0$ has no real solutions. We introduce the imaginary unit $I = \sqrt{-1}$ as solution to this equation. Consequently, the polynomial $x^2 + 1$ factors as $x^2 + 1 = (x - I)(x + I)$. Extending the field \mathbb{R} of real numbers with the imaginary unit gives rise to the field \mathbb{C} of complex numbers, as $\mathbb{C} = \{ a + bI \mid a, b \in \mathbb{R} \}$. We will see in this lecture how to extend this mechanism to any field and how to compute in Maple with algebraic numbers. Field extensions are very important in coding and cryptography.

4.1 Algebraic Numbers

We already encountered “modulo arithmetic”. For example,

```
[> 24 mod 7; # compute remainder of 24 after division by 7
[> -3 mod 7;
```

If we compute modulo 7, then we calculate in the finite field $\mathbb{Z}_7 = \{0, 1, 2, 3, 4, 5, 6\}$. This set is a finite subset of the integer numbers. Unlike the integer numbers, every element of \mathbb{Z}_7 has a multiplicative inverse, whence the name “field”. As example, we construct the multiplication table for \mathbb{Z}_7 . This table lists the results of all possible products of elements in \mathbb{Z}_7 . Contrary to the true nature of this course, we do some C style like programming, but see a good example of why we may want to joint the next two execution groups:

```
[> printf("* : 0 1 2 3 4 5 6\n");           # printing the elements of Z_7
[> for i from 0 to 6 do                       # multiplying every i between 0 and 6
>   printf("%d :",i);
>   for j from 0 to 6 do                     # with every j between 0 and 6
>     printf("%2d", i*j mod 7);
>   end do;
>   printf("\n");
[> end do;
```

Except for row 0 and column 0, we see that every row and column contains exactly one “1”. This means that in \mathbb{Z}_7 every element has a unique multiplicative inverse (unlike in \mathbb{Z}). For example,

```
[> 1/2 mod 7;
[> 2*4 mod 7;
```

We have seen the “Factor” command (observe the capital F!) to factor polynomials over finite fields:

```
[> p := x^2 + 3*x + 5;
[> Factor(p) mod 7;
[> Irreduc(p) mod 7;
```

We say that polynomial p is *irreducible over \mathbb{Z}_7* : it cannot be factored modulo 7.

Notice that p is reducible over \mathbb{Z}_3 :

```
[> Factor(p) mod 3;
```

It is easy to see that 1 is a root of p if we calculate modulo 3.

Returning to \mathbb{Z}_7 , we can declare a formal root of p .

```
[> alias(alpha = RootOf(p));
> y := subs(x=alpha,p);      # substitute x by alpha in p
```

Since Maple does not simplify automatically, we have to ask it explicitly to verify that alpha is a root of p.

```
[> evala(y);                # evaluate in algebraic number field
```

Algebraic numbers are numbers of the form $a + b\alpha$, where alpha is a root of an irreducible polynomial over the field containing a and b . Observe that α^2 simplifies to a linear expression in α :

```
[> evala(alpha^2) mod 7;
```

In our example we have $\mathbb{Z}_7(\alpha) = \{ a + b\alpha \mid a, b \in \mathbb{Z}_7 \}$. We can compute with algebraic numbers as we compute with ordinary numbers:

```
[> n1 := 2 + alpha: n2 := 3 + 2*alpha: m := n1*n2;
> Expand(m) mod 7;
```

Every algebraic number in a field extension has a multiplicative inverse:

```
[> invn1 := Expand(1/n1) mod 7;
> invn1*n1 = evala(invn1*n1) mod 7;
```

We have extended the finite field \mathbb{Z}_7 to the algebraic number field $\mathbb{Z}_7(\alpha)$. To compute in $\mathbb{Z}_7(\alpha)$ we use the command **evala**.

4.2 Complex Numbers

Complex numbers are a special kind of algebraic numbers.

```
[> I^2;
```

The letter I is Maple's notation for the imaginary unit. The above calculation shows that I is a root of $x^2 + 1$. In Maple we have extended the field of rational numbers with the root of $x^2 + 1$. For complex numbers, Maple offers some built-in functions:

```
[> z := 3 + 9*I;
> Re(z); Im(z);          # real and imaginary part
> abs(z); argument(z); # modulus and argument
```

Every complex number has a polar representation:

```
[> pz := abs(z)*exp(I*argument(z));
```

For general algebraic numbers we used **evala** a lot. For complex numbers, we have the **evalc** (**evaluate complex**) command:

```
[> sin(z);                # does not evaluate
> evalc(sin(z));          # evaluate symbolically over complex field
```

Be aware that complex functions are multivalued.

```
[> sqrt(x^2);
> simplify(%);
```

Maple knows that it would be wrong to simplify $\sqrt{x^2}$ to x , — in computer algebra, this used to be “the square root bug” (see [4]) — because it depends on the sign of the number. For example:

```
[> a := (-1+I)^2; b := (1-I)^2;
[> sqrt(a); sqrt(b);
```

The square root function is multivalued for complex numbers. The command `sqrt` does not recognize this, but we have an alternative in Maple, using the `Indexed RootOf`:

```
[> r1 := RootOf(x^2 - a, index = 1); # first branch of x^(1/2)
[> r2 := RootOf(x^2 - a, index = 2); # second branch of x^(1/2)
[> evalf(r1); evalf(r2); # see the two roots
```

This holds in general: $x^n - c = 0$ has n complex roots.

A primer on complex variables (with the issues most relevant to Maple) is in [2, Appendix A]. See [1] for a list of problems to test the capabilities of computer algebra system regarding complex analysis, adapted from [4].

4.3 Assignments

1. Generate the multiplication table for \mathbb{Z}_{12} . Which elements of \mathbb{Z}_{12} do have a multiplicative inverse?
2. With the operator `&^` we can define huge numbers without computing and storing their entire decimal expansion. For example, `a := 2&^3187 - 1` defines a large number of more than 3000 bits long. Maple can easily compute `a mod 7` for example. What are the ten last decimal places of `a`?
3. Use Maple to show that the polynomial $p := x^4 + 3x + 4$ is irreducible over $\mathbb{Z}_5 = \{0, 1, 2, 3, 4\}$. Declare α to be a root of p and express α^{13} as a polynomial in α of degree < 4 .
4. An irreducible polynomial in a finite field $\mathbb{Z}_p[x]$ is called “primitive”. Use the command `Nextprime` to find all primitive polynomials of degree 2 with coefficients in \mathbb{Z}_3 .
5. A set of numbers defines a field if every nonzero number has a multiplicative inverse. Show that \mathbb{Z}_4 is not a field. Extend \mathbb{Z}_2 with a formal root α of the polynomial $x^2 + x + 1$. Show that $\mathbb{Z}_2(\alpha)$ is a field with four elements.
6. The complex number z in polar representation is given by the radius (absolute value) $r = 3$ and angle (argument) $\theta = \pi/3$. Use Maple to find the exact (no floating-point) value of z in the form $a + bI$.
7. How can you show Maple knows the identity $e^{I\theta} = \cos(\theta) + \sin(\theta)I$ for any θ ?
8. Execute `solve(x^3-5)`; and show that all solutions of $x^3 - 5 = 0$ are of the form $\sqrt[3]{5}e^{I\theta}$, with θ being either 0 or $\pm\frac{2}{3}\pi$.
9. Take the complex number $z = 1+I$ and let Maple compute $\sqrt{1/z}$ and $1/\sqrt{z}$. Are the results symbolically the same? Are the results numerically the same? Give reasons for your answers, illustrated with the appropriate Maple instructions.
10. Maple has the symbol ∞ . Give examples of calculations with ∞ which are well defined. Also give an example of a calculation with ∞ which returns `undefined`.

References

- [1] H. Aslaksen. Can your computer do complex analysis? In M.J. Wester, editor, *Computer algebra systems: a practical guide*, pages 73–78. Wiley, 1999.
- [2] R.M. Corless. *Essential Maple 7. An introduction for Scientific Programmers*. Springer-Verlag, 2002.
- [3] A. Heck. *Introduction to Maple*. Springer-Verlag, third edition, 2003.
- [4] D.R. Stoutemyer. Crimes and misdemeanors in the computer algebra trade. *Notices of the American Mathematical Society*, 38(7):778–785, 1991.