

Maple for Math Majors

Roger Kraft

Department of Mathematics, Computer Science, and Statistics

Purdue University Calumet

roger@calumet.purdue.edu

9. Maple as a Numeric Calculator

- 9.1. Introduction

In this section we consider the difference between exact symbolic representations of numbers and decimal approximations of numbers. We show how Maple works with decimal approximations of numbers, we give examples of why we need decimal approximations, and we give some examples of how we can calculate decimal approximations of exact solutions to some problems.

[>

- 9.2. Symbolic vs. numeric arithmetic

Maple works much like a calculator, but it does a great deal more than a typical calculator. The biggest difference between Maple and a typical calculator is that Maple makes a distinction between working "symbolically" and working "numerically". Most hand held calculators cannot make this distinction, but this distinction is extremely important not just to Maple but to all of mathematics. This distinction has a lot to do with the decimal representation of numbers. With most hand held calculators, if you enter the number $1/3$, the calculator will display $.333333333$ (the number of threes depends on the calculator). It is important to know that $1/3$ is not equal to $.333333333$ for any finite number of threes, but a hand held calculator has no other choice. On the other hand, Maple gives you a choice of working with $1/3$ either symbolically or numerically. Let us demonstrate the difference.

When you enter the number $1/3$ into Maple, by default it treats this as a symbolic expression for one divided by three; it does not treat it as a decimal number.

[> **$1/3;$**

If we want Maple to treat $1/3$ as a decimal number, we can put a decimal point somewhere in the $1/3$ (it does not matter where).

[> **$1./3;$ $1/3.;$ # Notice the decimal points.**

Here is an example of how Maple can do "symbolic" arithmetic.

[> **$1/3 + 1/3;$**

If we put a decimal point anywhere in the last command, then Maple will do a decimal (or numeric) calculation. For example

[> **$1./3 + 1/3;$ $1/3 + 1/3.;$ # Notice the decimal points.**

We know that $1/3$ plus $1/3$ is $2/3$, but let us check this by asking Maple for the decimal version of $2/3$.

```
[ > 2/3.; # Again, notice the decimal point.
```

Look carefully at the last two results. Is there something wrong with Maple? $1/3$ plus $1/3$ was not the same as $2/3$ when we did the calculation numerically. This shows us two things. First, there really is a difference in Maple between doing a symbolic calculation and doing a numeric calculation. Secondly, even when Maple does a numeric calculation it does not quite work like a hand held calculator; every hand held calculator in the world will tell you that $1/3$ plus $1/3$ is $2/3$.

Let us try another example. Let us calculate the square root of two. Here is how we enter the square root of two into Maple.

```
[ > sqrt(2);
```

Maple treats square root of two as a symbolic representation for the number whose square is two. (Notice how the result does not look like the command. The command is in a form that Maple refers to as **Maple notation**, and the result is in a form that Maple refers to as **Standard Math notation**.) If we want Maple to treat the square root of two as a decimal number, we put a decimal point in the command.

```
[ > sqrt(2.);
```

The square root of two squared should be 2. Let us check, first symbolically, then numerically.

```
[ > sqrt(2) * sqrt(2);
```

```
[ > sqrt(2.) * sqrt(2.); # Notice the decimal points.
```

Unfortunately, square root of two squared was not exactly two when computed numerically.

Let us show that even very small errors can matter if we get a lot of them and they accumulate.

```
[ > 1./3 + 1./3 - 2./3;
```

```
[ > add( 1./3 + 1./3 - 2./3, i=1..1000000);
```

```
[ >
```

The `evalf` function is not linear.

```
[ > evalf(1/3) + evalf(1/3) - evalf(1/3 + 1/3);
```

```
[ >
```

9.3. Decimal places, more decimal places, and correct decimal places

```
[ >
```

9.4. Online help for numeric calculations

```
[ >
```