February 16, 2000

Experimental mathematics

Some Maple instructions

In this assignment we will explore how rapidly `Maple` can solve some simple modular equations. Remember: you can't break the program – you can always exit it and try again if something goes wrong. You should be at a terminal using `Maple` when you read the remainder of what is written here. Two new instructions will be needed, and trying them as you read this will almost certainly improve your understanding.

THE FIRST INSTRUCTION *msolve*

msolve solves modular equations. Suppose you want to solve

$$7x = 3 \mod 13.$$

Type the instruction *msolve(7*x=3, 13);* (remember that you must end an instruction with a semicolon so that `Maple` will know you're done telling it what to do). The answer should be *{x=6}* and I hope you realize that $7 \cdot 6 = 42$, and that 42 is $(3 \cdot 13) + 3$, so the remainder of 42 after division by 13 is 3.

Now go to the keyboard and be sure you know the answers to these questions:

- Type *msolve(2*x=4,8);* and explain the result.
- Type *msolve(2^x=1,8);* and explain the result. ^ is exponentiation, not multiplication.

Remember you can type *help(msolve);* if you want information about *msolve*.

THE SECOND INSTRUCTION *showtime*

This instruction will allow you to see how much computer processing time `Maple` uses. Type the two instructions *readlib(showtime); showtime(on);* and see what happens.

The first instruction tells `Maple` to load the command *showtime* and the second turns *showtime* on. In order to tell you that *showtime* is on, the `Maple` prompt `[>` has changed to **[01 :=**. Now type *3333^4444;* and look at the result. First, you should get lots and lots of output (3333 multiplied 4444 times with itself is quite a big number as you can see) – don't worry. The output should end with the digits *81521*. Then *showtime* will report the time and computer "space" used by the computation. I'm checking `Maple` on three computer systems as I type this: on one, the instruction took 3.69 seconds, on another, it took 0.29 seconds, and on the third, 0.04 seconds. Computers run at different speeds. If you want, you can type *help(showtime);* to see what showtime does and what it reports.

Your assignment

This will be a purely empirical* study of how fast `Maple` solves some modular equations. You will need to use *showtime* and *msolve* to do this assignment. Please record some observations, try to see patterns and make some guesses.

---

* A dictionary's definition of this word begins: "based or acting on observation or experiment, not on theory."

Choose some **4**-digit odd number. How long does `Maple` take to solve the two equations $2x = 1$ and $2^x = 1$ mod the odd number you chose?* Do this again for a **8**-digit odd number. And again for a **12**-digit odd number, and again ... Repeat these instructions for a bit, increasing the number of the digits each time by 4. Report the timing results to me.

For instance, I chose my first number to be 2335. I typed *msolve(2*x=1,2335);* and got the answer $\{x = 1168\}$ along with a line reading `time = 0.01, bytes = 22462`. Then I typed *msolve(2^x=1,2335);* and got the answer $\{x = 932\_NN1\sim\}$ (which means that every integer multiple of 932 is also a solution to that equation) along with a line reading `time = 0.15, bytes = 1183898`. Below is a table in which you can fill in the times you get. As an example, I have already filled in what I got for my 4 digit number, 2335.

One useful format for the report would be to draw a bar graph or a line graph of the results (on the same axes) of the two series of experiments in a pair of contrasting colors. The horizontal axis of the graph should be the number of digits. The vertical axis should be the time taken by `Maple` to find the solutions. In addition, give a very brief written qualitative description of the graphs: explain how difficult `Maple` finds solving the two series of equations as the number of digits increases.

I tried this on `Eden`'s version of `Maple` recently. I began to get embarrassed at around 30-35 digits – very likely you will see why. You may be able to push things to about 40 digits, but don't do too much, or we'll all get in trouble for trying this "exercise"!

# of digits in the modulus	**Time to solve 2x = 1 mod n**	**Time to solve 2^x = 1 mod n**
4	0.01	0.15

Note S. Greenfield wrote a first draft of this assignment. It was then edited and improved by S. Radimirovic, who knows quite a lot more about `Maple`.

* Why should the modulus be odd? How many solutions would you expect if the modulus were even?