

For each error herein, you get extra-credit points for being the first to report it by e-mail.

1 Pine Mailer Not Compulsory

You don't *have* to use the pine mailer to send stuff. It is certainly OK to use any mailer that will get the job done.

2 Email needs a SUBJECT

Whatever mailer you use, include a subject line, preferably mentioning MATH 171. You don't want your stuff mistaken for SPAM.

3 More Advanced Book

Introduction to Maple by André Heck
QA155.7.E4 H43 1993 in the Albertsons Library.

4 Notes on Reading Chapter 2 for Thursday, 9/15/05

The index of our text is pretty good.

We want you to get with the ideas of chapter 2. Some of the suggestions below lead to Maple errors. You need to be able to recognize when this is happening (“gigo”).

- (1) Maple expressions
 - (a) Syntax or grammar.
 - (b) How to assign one to a variable: `exprf := x + Pi*x^2;` assigns an expression to the variable `exprf`.
 - (c) Maple's value for `exprf(2);`?
 - (d) Then how *does* one evaluate the expression `exprf` for the value $x = 2$? (check out page 18)
 - (e) Maple's value for `2*x^3 - exprf;`?
 - (f) Plotting one-variable expressions.
 - (i) Outcome of `plot(exprf)?`

- (ii) Outcome of `plot(exprf,x)`?
- (iii) Outcome of `plot(exprf,x=-Pi/2..0)`?
- (iv) How does one crop the plot vertically?

(2) Maple functions

- (a) Syntax or grammar: “arrow defining” a Maple function.
- (b) How to assign one to a variable: `funf := x -> 4*x - x^2`; defines and names the function `funf`.
- (c) “Arrow Defining” a function of two or more variables: `snarf := (a,b,c) -> b^2 - 4*a*c`;
- (d) Maple’s value for `funf(2)`; (applying the function `funf` to the value `2`)?
- (e) Maple’s value for `funf(W)`;
- (f) Maple’s value for `funf(t + h)`;
- (g) Maple’s value for `2*x^3 - funf`;
- (h) Maple’s value for `2*x^3 - funf(2*x)`;
- (i) Maple’s value for `snarf(1,-2,1)`?
- (j) Plotting one-variable functions.
 - (i) Check into `unapply` on page 20.
 - (ii) Outcome of `plot(funf)`?
 - (iii) Outcome of `plot(arctan)`?
 - (iv) Outcome of `plot(exprf,-Pi..0)`?
 - (v) Outcome of `plot(funf(x))`?
 - (vi) Outcome of `plot(t -> 5 - 2*t, 0..3)`?
 - (vii) How does one crop the plot vertically?

(3) Maple equations

- (a) Syntax or grammar: `3*x + 2*y = 5*x - 3*y`;
- (b) How to assign one to a variable: `EQzz := 3*x + 2*y = 5*x - 3*y`;
- (c) Do you understand what `lhs(EQzz)` does?
- (d) `rhs(EQzz)`?
- (e) `solve(EQzz, x)`;
- (f) `EQ321 := (x - 2)^2 + 2*(y+3)^2 = 4`;
- (g) Plotting equations.
 - (i) `with(plots)`; or `with(plots, implicitplot)`;
 - (ii) `implicitplot(EQ321, x=-10..10, y= -10..10)`;

- (iii) Option `numpoints=10000`
- (iv) Option `scaling=CONSTRAINED`

(4) Composite plots

- (a) Using square brackets to plot several functions over the same domain.
- (b) Pasting together plots from disparate sources:
 - (i) Assigning plot structures to variables (page 23). Check out
 - `plot(cos);`
 - `AA := plot(sin);`
 - `AAA := plot(cos);`
 - (ii) `with(plots);` or `with(plots, display);`
 - (iii) `display([AA,AAA]);`

(5) Hard-Core Equation Solving

- (a) Here's a sequence of commands

```
EQ := 7 = 3^(2*x - 4);
solve(EQ, x);
expand(%);
evalf(%);
```

- (b) Repeat the above sequence with a different definition of the equation:

```
EQ := (2*x - 3)^2 = 37;
```

The outcome is not altogether satisfactory, right? One way to get Maple to pay attention to both roots is to surround the `solve` command with square brackets (put the semicolon in the right place). There's another suggestion on page 27 of the text.

- (c) Here's another fun command sequence:

```
EQ := (2*x+5)*(2*x-5)*(3*x-7)=0; [solve(EQ, x)];
sort(%);
evalf(%);
```

- (d) Approximate solutions: what we often have to settle for:

```
EQ := sin(3*x) = exp(x/10)/3;
[solve(EQ, x)];
evalf(%);
RHS_0 := lhs(EQ) - rhs(EQ);
plot(RHS_0, x);
plot(RHS_0, x=-1..1);
Least_Positive_Root := fsolve(RHS_0 = 0, x=0..1/5);
```

5 Assignment #2

Page 34: 10, 11, 12, , 13, 14, 16c, 17a, 20.