Outlines, for Mathematical Computing course in Maple.

**Preface:**

The following course outline is intended to be like a detailed contents page, directing you to the relevant part of the notes, and reminding you of the topics covered in each chapter.

**Acknowledgements:**

Francis Wright, Tim Morgan.

**Last edited:**

September 2015, by VK.

# Chapter 1 Outline – *Basics*

* **Parts** of Maple
* **Documents** vs **Worksheets**
* **Math** **mode**
* **Text mode**/Writing
* Basic operations (**multiplication**, fractions, indices, **two-dimensional layout**)
* **Palettes**
* **Shortcuts**
* **Printing** and Exporting documents
* Maple **Help**
* **Error** Messages

# Chapter 2 Outline – *More Basics and Important Functions*

Maple Basics

* **Execution Groups**
* **Output Labels**
* Maple **worksheet** vs Maple **Document**

Common Usage

* ***π***and***e***
* **Inequality** Symbols

Functions

* **Naming** Functions
* **Common** Functions
* **Operators**
* **diff()** and **int()**
* **op()**
* **whattype()**
* **Floating Point Approximation. evalf()**

# Chapter 3 Outline – *More Functions*

* **sum()**, **product()**
* **add()**, **mul()**
* **eval()**
* **limit()**
* **taylor()**, **series()**
* **Inert** functions
* **value()**
* **Subscripts**
* **evalf()** and more on **approximations**
* Simplifying with **simplify()**, **expand()**, **factor()**
* **Prime** number functions
* **isprime()**
* **nextprime()**
* **prevprime()**
* **ithprime()**
* **ifactor()**

# Chapter 4 Outline – *Sets, Assignments, Lists, Sequences*

Assignments

* Maple Names
* **Assignment** Operator and **Removing** assignments
* **Restart** Maple **server**

Sets

* Set **membership**
* **Empty** set
* **Subsets**, Proper subsets
* **Power** sets
* **Union** and intersection (**intersect**)
* **Set** **difference**, **Symmetric** **difference**
* **nops()**, **cardinality**

Lists and Sequences

* **seq()** and **$**
* Generating nested structures with seq() and $

Other

* **Predicates, is()**
* **Binomial** Coefficient

# Chapter 5 Outline – *Plotting*

* **strings**, **cat()**
* **2-D plots**
* Using **Smart Plot**, or using **plot()**
* Plotting **Points**, **Line segments**, Geometrical Plotting
* **Multiple Plots**
* **Implicit Plots**
* **Assigning** **Labels** to Plots
* **Packages** and **with()**
* the **plots** package
* **display()**
* **3-D Plots**

# Chapter 6 Outline – *Boolean Algebra and Boolean Evaluation*

* **Boolean constants**
* **Relational operators**
* **lhs()**, **rhs()**
* **evalb()**, **is()**
* Boolean **operators**
* **and**, **or**, **not**, **xor**, **implies**
* using **operators** as **functions**
* Boolean simplification, evaluation
* Aside: Testing if a set of sets is **pairwise disjoint**
* Aside: Link between **sets and logic**

# Chapter 7 Outline – *Control Structures*

* Grouping **multi-statements**, **terminators**, (**:**, **;**)
* **Emergency Stops**
* **print()**
* **Comments**
* **Parallel Assignment**
* **Control Structures**
* **for**… **in**/**∈**… **from**… **by**… **to**… ***do***… ***end do***
* **if**… **then**
* **next**, **break**
* **while**
* Accessing Data structures using **Indexing**
* **Generating sequences, sums, products**, with a **loop**
* Aside: Multi-step **recursive** **sequences**

# Chapter 8 Outline – *Functions*

* Defining **Functions** in maple
* **Map** and **Tilde** (~)
* **Select**, **Remove**, **Selectremove**
* Aside: Mapping vs Expression.
* Finding range of a function with finite domain
* Multidimensional sets
* **Piecewise functions**
* Defining finite functions using:
* table or list
* **parallel** **assignment**
* **assign()**
* **Plotting** functions on finite sets
* Aside: Testing function equality
* **Injectivity**, **Surjectivity**, **Bijectivity** checks (on finite set)
* Constructing the **inverse** function on a finite set. (using a loop or using assign)

# Chapter 9 outline – *Relations, Solving equations, Division*

Relations

* Inert vs active predicates/relations
* Defining **Relations**
* Neutral operator (&)
* Testing **reflexive**, **symmetric**, and **transitive** properties of a relation
* Testing a **partition**.
* **Equivalence relations** from **Partitions** and Vice Versa.

Solving equations

* Solving equations with **solve()**.
* i.e. polynomials, multiple variables, inequalities, simultaneous equations
* various arguments: ‘**explicit**’ and ‘**allsolutions**’
* Numerical approximations to equation solutions using **fsolve()**.
* various arguments: ‘**complex**’, isolating intervals

Division

* **numer()** and **denom()**
* **iquo()** and **irem()**
* **igcd()** and **ilcm()**

# Chapter 10 Outline – *Procedures*

* **Procedure** definition
* **Local** and **Global** **Variables**
* **return**
* **Recursive** **procedures**
* **Tracing** and **debugging** procedures
* Tips on writing procedures well
* see main lecture notes
* e.g. doublefactorial()
* Aside: Computing permutations recursively
* see main lecture notes

# Chapter 11 Outline – *Complex Numbers and complexplot()*

* Complex Numbers, **Imaginary unit**
* Fundamental **Complex Number Functions**
* **Simplifying** complex numbers
* **Numerical approximation**
* Real vs complex indeterminates
* **evalc()**
* **complexplot()**
* Aside: De Moivre’s Theorem