

Programming for Engineers with *Mathematica*

Course Notes

Table of Contents

1. Introduction

1.1 Introduction to the Course

- Course aims
- Assessment
- How to learn to program and do well on the tests
- Class schedule

1.2 Introduction to Programming Languages

- Objectives
- What is programming?
- What is a programming language?
- The level of a programming language
- Types of programming languages
- Other important languages

1.3 Introduction to Mathematica

- What is Mathematica?
- Rationale for adopting Mathematica's language

1.4 References and Resources

- Recommended text
- Bibliography
- Wolfram Research

1.5 Getting Started

- The Front End and the Kernel
- Notebooks

- Input
- When you do not get what you expect
- Basic Syntax
- Cells
- Referring to previous output
- Getting help

1.6 Exercise: A Tour of Mathematica

- Touring Mathematica in the Help Browser
- Introductory topics
- Important but not introductory topics

2. Expressions and Evaluation

2.1 Introduction

2.2 Expressions

- Objectives
- What is an expression?
- But not everything looks like an expression!
- Ways of looking at expressions
- Finding the **Length** of an expression
- Exercises

2.3 Evaluation

- Objectives
- SMathematica evaluates expressions
- Holding evaluation
- Exercises

2.4 Assignment

- Objectives
- Using **Set**
- Using **SetDelayed**
- Exercises

2.5 Lists and Tables

- Objectives
- Lists
- Listability
- Algebraic and calculus operations on lists
- Constructing lists with **Table**
- Efficiency in list manipulation
- Exercises

2.6 Problems

- Problem 1: Exploring the validity of expressions
- Problem 2: Exploring the evaluation sequence with **Unique**
- Problem 3: Exploring the difference between **Set** and **SetDelayed**
- Problem 4: Exploring expressions with **FullForm**
- Problem 5: Generating data with **Table**

3. Rules and Patterns

3.1 Introduction

3.2 Rules

- Objectives
- Using **Rule** and **ReplaceAll**
- Using **RuleDelayed**
- Exercises

3.3 Patterns

- Objectives
- What is a pattern?
- Named patterns
- Exercises

3.4 Compound Expressions

- Objectives
- Using compound expressions
- Exercises

3.5 Problems

- Problem 1: Applying rules to expressions
- Problem 2: Applying rules to formulas
- Problem 3: Applying rules to data
- Problem 4: Applying rules and conditions to data

4. Writing Functions

4.1 Introduction

- Objectives

4.2 Understanding Function Definitions

- The need for patterns
- The need for delayed assignment

4.3 Clearing Function Definitions

- Why it is important to clear old function definitions
- How to ensure you don't go crazy

4.4 An Example with Three Variables

- Defining the load capacity of a thrust bearing
- Using the formula for the load capacity

4.5 Summary: The Form of a Function Definition

- Summary
- Exercises

4.6 Problems

- Problem 1: Computing amplitudes of vibration
- Problem 2: Reflecting graphs

Problem 3: Rotating graphs
Problem 4: Checking algorithms
Problem 5: Calculating data averages

5. Writing Modules

5.1 Introduction

Objectives

5.2 The Need for Modules

The problem of writing functions with several steps
The solution

5.3 A Template for Using Module

The structure of a module
Example of a module

5.4 How to Develop a Module

Seven steps
Example problem using a module
Exercises

5.5 Advanced Concept: How Module Keeps Values Local

5.6 Problems

Problem 1: Manipulating data
Problem 2: Creating plots
Problem 3: Playing sound

6. Functional Programming

6.1 Introduction

What is functional programming?
Objectives

6.2 Map

Mapping functions over lists
Keeping your code general
Mapping functions over expressions
Shorthand for **Map**
Exercises

6.3 Apply, Thread and Outer

Apply
Exercise
Thread
Exercises
Outer
Exercise

6.4 Anonymous Functions

Defining a function without giving it a name
Exercise

6.5 Nest, NestList and FixedPoint

Nest

NestList

Exercise

FixedPoint

Exercises

6.6 Summary: When to Use Functional Programming

Don't dismember lists

Map, Apply, Thread and **Outer**

Anonymous functions

Nest, NestList and **FixedPoint**

6.7 Problems

Problem 1: Calculating the centre of gravity

Problem 2: Smoothing data

Problem 3: Exploring vehicle suspension dynamics

Problem 4: Discovering repeated data

Problem 5: Distances from a fixed point

Problem 6: Newton's method for any number of variables (More difficult!)

7. Rule Based Programming

7.1 Introduction

What is a rule-based program?

Objectives

7.2 Constraining Rules with Simple Patterned Arguments

Rules with different number of arguments

Rules with **List** arguments

Rules with other argument patterns

Rules with constant arguments

The basic concept of rule-based programming

Conflicting rules

Exercises

7.3 Constraining Rules with **x_Head**

Types of argument

Using the **Head** of an argument

Exercise

7.4 Constraining Rules through Predicates

Built-in predicates

Other predicate forms

Boolean operations

Using built-in predicates to constrain functions

Exercises

7.5 Constraining Rules through Conditions

Using predicates which impose a condition on several arguments
Exercises

7.6 Problems

Problem 1: Creating a gear selector program
Problem 2: Creating a unit conversion program
Problem 3: Writing functions which includes special cases
Problem 4: A more challenging program

8. Graphics Programming

8.1 Introduction

What is graphics programming?
Objectives

8.2 Showing Graphics Objects

The plot as a side-effect
Showing plots
Exercise

8.3 The FullForm of a Graphics Object Expression

The internal representation of a graphics object
Extracting the points of a graphics object
Exercises

8.4 Two Dimensional Graphics Primitives

Defining graphics primitives
Exercises

8.5 Graphics Options

Extracting graphics options
Setting graphics options
Modifying options with **Show**
Exercises
AbsoluteOptions
Exercise

8.6 Style Directives

Using graphics style directives
Color
Exercises
Point size
Exercise
Thickness
Exercises

8.7 Controlling Graphic Output

Graphic displays as side-effects
Exercise

8.8 Graphics Packages

Exploring graphics packages
Exercises

8.9 Graphics Arrays

Showing an array of graphics images
Exercise

8.10 Animation

Generating an animation sequence
Generating a spin sequence
Exercises

8.11 Problems

Problem 1: Modifying the points of a plot
Problem 2: Further modification of the points of a plot
Problem 3: Generating random graphics displays
Problem 4: Picking off graphics coordinates
Problem 5: Using color directives
Problem 6: Viewing plots from different angles
Problem 7: Generating graphics arrays
Problem 8: A function which generates a graphics array

9. Concluding Techniques

9.1 Introduction

What are these concluding techniques?
Objectives

9.2 Timing

Timing a computation
Exercises

9.3 Printing

Printing an intermediate result
Exercises

9.4 Advanced Patterns

Patterns of sequences of elements
Exercises

9.5 Timing

Recursive functions
Recursion Limits

9.6 Dynamic Programming

The concept of a dynamic program
Looking at the rule base for a recursively defined function

9.5 Procedural Programming

Traditional programming constructs
Do

**For
While****9.5 Problems**

- Problem 1: Exploring timing
- Problem 2: Exploring advanced patterns
- Problem 3: Exploring dynamic programming

A. Packaging Your Program

A.1 Introduction

- What are these concluding techniques?
- Objectives

A.2 Documenting a Program

- Documentation files
- Your Workbook
- Your Journal
- The Package
- The User Guide

A.3 Your Workbook

- Hints on keeping a Workbook
- Example: Developing a program in your workbook

A.4 Your Journal

- What is a Journal?
- The **Usage** statement
- The function definition
- The function implementation
- The function testing
- Example: Transferring a program to your Journal

A.5 The package

- Creating a package
- Loading a Package (on your own computer)
- Demonstration: Creating and loading a package (on your own computer)

A.5 The User Guide

- What is a User Guide for?
- What should a User Guide include?

A.6 A Final Note on the Art of Programming

- The big tip
- Exercises
- A final problem

B. Programming Project

B.1 Introduction

- The scenario
- The deliverables

B.2 Build Your Model

- The power balance model
- Driving power
- Traction power
- Road load power
- Excess power

B.3 Setting up Your Data

- Road data
- Car data
- Protecting your data symbols
- Making substitution rules

B.4 Exploring the Performance

- Calculating the power in terms of the car velocity
- Calculating the top speed
- Enhancing your plots
- Plotting power graphs
- Calculating the top speed on an incline
- Exploring your own interests

C. Past Test Questions

Test 1: Expressions, Evaluation, Rules, and Patterns

Test 2: Writing Functions

Test 3: Using Modules

Test 4: Functional Programming

Test 5: Rule-Based Programming

Test 6: Graphics Programming

Test 7: Concluding Techniques

Test 8: Packaging Your Program