

# Contents

Preface · page xi

## I An introduction to *Mathematica*

### 1.1 Overview of basic operations · 1

*Numerical and symbolic computation · Graphics and visualization · Working with data · Dynamic interactivity · Programming*

### 1.2 Getting started · 14

*Starting up Mathematica · The notebook interface · Entering input · Mathematical expressions · Syntax of functions · Lists · Semicolons · Alternative input syntax · Comments · Errors · Getting out of trouble · The front end and the kernel*

### 1.3 Getting help · 25

*Function information · The Documentation Center*

## 2 The *Mathematica* language

### 2.1 Expressions · 29

*Types of expressions · Atoms · Structure of expressions · Evaluation of expressions · Exercises*

### 2.2 Definitions · 40

*Defining variables and functions · Immediate vs. delayed assignments · Term rewriting · Functions with multiple definitions · Exercises*

### 2.3 Predicates and Boolean operations · 48

*Predicates · Relational and logical operators · Exercises*

### 2.4 Attributes · 53

*Exercises*

- 3 Lists
  - 3.1 Creating and displaying lists · 58  
*List structure and syntax · List construction · Displaying lists · Arrays · Exercises*
  - 3.2 The structure of lists · 67  
*Testing a list · Measuring lists · Exercises*
  - 3.3 Operations on lists · 70  
*Extracting elements · Rearranging lists · List component assignment · Multiple lists · Exercises*
  
- 4 Patterns and rules
  - 4.1 Patterns · 85  
*Blanks · Pattern matching by type · Structured patterns · Sequence pattern matching · Conditional pattern matching · Alternatives · Repeated patterns · Functions that use patterns · Exercises*
  - 4.2 Transformation rules · 102  
*Creating and using replacement rules · Example: counting coins · Example: closed paths · Example: finding maxima · Exercises*
  - 4.3 Examples and applications · 109  
*Finding subsequences · Sorting a list · Exercises*
  
- 5 Functional programming
  - 5.1 Introduction · 116
  - 5.2 Functions for manipulating expressions · 118  
*Map · Apply · Thread and MapThread · The Listable attribute · Inner and Outer · Select and Pick · Exercises*
  - 5.3 Iterating functions · 132  
*Nest · FixedPoint · NestWhile · Fold · Exercises*
  - 5.4 Programs as functions · 137  
*Building up programs · Example: shuffling cards · Compound functions · Exercises*
  - 5.5 Scoping constructs · 146  
*Localizing names: Module · Localizing values: Block · Localizing constants: With · Example: matrix manipulation · Exercises*
  - 5.6 Pure functions · 153  
*Syntax of pure functions · Using pure functions · Example: searching for attributes and options · Exercises*

- 5.7 Options and messages · 164  
*Options · Messages · Exercises*
- 5.8 Examples and applications · 170  
*Hamming distance · The Josephus problem · Regular graphs/polygons · Protein interaction networks · Palettes for project files · Operating on arrays · Exercises*
- 6 Procedural programming
  - 6.1 Loops and iteration · 190  
*Newton's method · Do loops and For loops · Example: random permutations · While loops · NestWhile and NestWhileList · Exercises*
  - 6.2 Flow control · 208  
*Conditional functions · Piecewise-defined functions · Which and Switch · Argument checking · Exercises*
  - 6.3 Examples and applications · 219  
*Classifying points · Sieve of Eratosthenes · Sorting algorithms · Exercises*
- 7 Recursion
  - 7.1 Fibonacci numbers · 231  
*Exercises*
  - 7.2 Thinking recursively · 234  
*Length of a list · Recursion with multiple arguments · Multiplying pairwise elements · Dealing cards, recursively · Finding maxima · Higher-order functions · Exercises*
  - 7.3 Dynamic programming · 239  
*Exercises*
  - 7.4 Classical examples · 244  
*Merge sort · Run-length encoding · Exercises*
- 8 Numerics
  - 8.1 Numbers in Mathematica · 251  
*Types of numbers · Digits and number bases · Random numbers · Exercises*
  - 8.2 Numerical computation · 265  
*Precision and accuracy · Representation of approximate numbers · Exact vs. approximate numbers · High precision vs. machine precision · Computations with mixed number types · Working with precision and accuracy · Exercises*

8.3 Arrays of numbers · 282  
*Sparse arrays · Packed arrays · Exercises*

8.4 Examples and applications · 291  
*Newton's method revisited · Radius of gyration of a random walk · Statistical tests · Exercises*

## 9 Strings

9.1 Structure and syntax · 310  
*Character codes · Sorting lists of characters · Ordered words · Exercises*

9.2 Operations on strings · 316  
*Basic string operations · Strings vs. lists · Encoding text · Indexed symbols · Anagrams · Exercises*

9.3 String patterns · 325  
*Finding subsequences with strings · Alternatives · Exercises*

9.4 Regular expressions · 332  
*Word stemming · Exercises*

9.5 Examples and applications · 343  
*Random strings · Partitioning strings · Adler checksum · Search for substrings · DNA sequence analysis · Displaying DNA sequences · Blagrams · Exercises*

## 10 Graphics and visualization

10.1 Structure of graphics · 365  
*Graphics primitives · Graphics directives · Graphics options · Combining graphics elements · Structure of built-in graphics functions · Example: Bézier curves · Example: hypocycloids · Exercises*

10.2 Efficient structures · 386  
*Multi-objects · GraphicsComplex · Numeric vs. symbolic expressions · Exercises*

10.3 Sound · 396  
*The sound of mathematics · Sound primitives and directives · Exercises*

10.4 Examples and applications · 402  
*Space filling plots · Plotting lines in space · Simple closed paths · Points in a polygon · Visualizing standard deviations · Root plotting · Trend plots · Brownian music · Exercises*

## 11 Dynamic expressions

11.1 Manipulating expressions · 449  
*Control objects · Control wrapper · Viewers · Animating the hypocycloid · Visualizing logical operators · Exercises*

## 11.2 The structure of dynamic expressions · 470

*Dynamic · DynamicModule · Dynamic tips · Exercises*

## 11.3 Examples and applications · 481

*Creating interfaces for visualizing data · File openers · Dynamic random walks · Apollonius' circle · Exercises*

## 12 Optimizing Mathematica programs

### 12.1 Measuring efficiency · 494

*Evaluation time · Memory storage*

### 12.2 Efficient programs · 496

*Low-level vs. high-level functions · Pattern matching · Reducing size of computation · Symbolic vs. numeric computation · Listability · Pure functions · Packed arrays · Exercises*

### 12.3 Parallel processing · 515

*Basic examples · Distributing definitions across subkernels · Profiling · Exercises*

### 12.4 Compiling · 523

*Compile · Compiling to C · Exercises*

## 13 Applications and packages

### 13.1 Random walk application · 534

*Lattice walks · Off-lattice walks · RandomWalk · Error and usage messages · Visualization · Animation · Exercises*

### 13.2 Overview of packages · 555

*Working with packages · Package location*

### 13.3 Contexts · 558

### 13.4 Creating packages · 563

*Package framework · Creating and installing the package · RandomWalks package · Running the package · Exercises*

## Solutions to exercises

- 2 The *Mathematica* language · 575
- 3 Lists · 578
- 4 Patterns and rules · 582
- 5 Functional programming · 588
- 6 Procedural programming · 614
- 7 Recursion · 621
- 8 Numerics · 626
- 9 Strings · 638
- 10 Graphics and visualization · 651
- 11 Dynamic expressions · 666
- 12 Optimizing *Mathematica* programs · 676
- 13 Applications and packages · 681

Bibliography · 687

Index · 695