1 Getting Started

1.1 A Brief Introduction to C++
1.1.1 C++ is “Object-Oriented”
1.1.2 Why You Should Write Scientific Programs in C++
1.1.3 Why You Should Not Write Scientific Programs in C++
1.1.4 Scope of This Book

1.2 A First C++ Program

1.3 Compiling a C++ Program
1.3.1 Integrated Development Environments
1.3.2 Compiling at the Command Line
1.3.3 Compiler Flags

1.4 Variables
1.4.1 Basic Numerical Variables
1.4.2 Other Numerical Variables
1.4.3 Mathematical Operations on Numerical Variables
1.4.4 Division of Integers
1.4.5 Arrays
1.4.6 ASCII Characters
1.4.7 Boolean Variables
1.4.8 Strings

1.5 Simple Input and Output
1.5.1 Basic Console Output
1.5.2 Keyboard Input

1.6 The assert Statement

1.7 Tips: Debugging Code

1.8 Exercises

2 Flow of Control

2.1 The if Statement
2.1.1 A Single if Statement
2.1.2 Example: Code for a Single if Statement
2.1.3 if–else Statements
2.1.4 Multiple if Statements
2.1.5 Nested if Statements
2.1.6 Boolean Variables ........................................... 28
2.2 Logical and Relational Operators ............................. 29
2.3 The while Statement ........................................... 30
2.4 Loops Using the for Statement ............................... 32
  2.4.1 Example: Calculating the Scalar Product of Two Vectors .. 34
2.5 The switch Statement .......................................... 34
2.6 Tips: Loops and Branches .................................... 35
  2.6.1 Tip 1: A Common Novice Coding Error .................. 35
  2.6.2 Tip 2: Counting from Zero ............................. 36
  2.6.3 Tip 3: Equality Versus Assignment ....................... 37
  2.6.4 Tip 4: Never Ending while Loops ......................... 38
  2.6.5 Tip 5: Comparing Two Floating Point Numbers .......... 39
2.7 Exercises ....................................................... 39

3 File Input and Output ........................................... 43
  3.1 Redirecting Console Output to File ......................... 43
  3.2 Writing to File ............................................. 44
    3.2.1 Setting the Precision of the Output ................... 46
  3.3 Reading from File ........................................... 47
  3.4 Reading from the Command Line ............................. 49
  3.5 Tips: Controlling Output Format ........................... 50
  3.6 Exercises ................................................... 51

4 Pointers ........................................................ 55
  4.1 Pointers and the Computer's Memory ....................... 55
    4.1.1 Addresses ............................................. 55
    4.1.2 Pointer Variables ................................... 56
    4.1.3 Example Use of Pointers .............................. 56
    4.1.4 Warnings on the Use of Pointers ...................... 57
  4.2 Dynamic Allocation of Memory for Arrays .................. 58
    4.2.1 Vectors ............................................... 59
    4.2.2 Matrices .............................................. 60
    4.2.3 Irregularly Sized Matrices ........................... 61
  4.3 Tips: Pointers .............................................. 62
    4.3.1 Tip 1: Pointer Aliasing ............................. 62
    4.3.2 Tip 2: Safe Dynamic Allocation ....................... 63
    4.3.3 Tip 3: Every new Has a delete ......................... 63
  4.4 Exercises ................................................... 64

5 Blocks, Functions and Reference Variables .................... 65
  5.1 Blocks ...................................................... 65
  5.2 Functions .................................................. 66
    5.2.1 Simple Functions ..................................... 66
    5.2.2 Returning Pointer Variables from a Function .......... 69
    5.2.3 Use of Pointers as Function Arguments ................ 70
    5.2.4 Sending Arrays to Functions .......................... 71
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.2</td>
<td>Templates for Polymorphism</td>
<td>133</td>
</tr>
<tr>
<td>8.3</td>
<td>A Brief Survey of the Standard Template Library</td>
<td>134</td>
</tr>
<tr>
<td>8.3.1</td>
<td>Vectors</td>
<td>134</td>
</tr>
<tr>
<td>8.3.2</td>
<td>Sets</td>
<td>137</td>
</tr>
<tr>
<td>8.4</td>
<td>Tips: Template Compilation</td>
<td>139</td>
</tr>
<tr>
<td>8.5</td>
<td>Exercises</td>
<td>140</td>
</tr>
<tr>
<td>9</td>
<td>Errors and Exceptions</td>
<td>141</td>
</tr>
<tr>
<td>9.1</td>
<td>Preconditions</td>
<td>142</td>
</tr>
<tr>
<td>9.1.1</td>
<td>Example: Two Implementations of a Graphics Function</td>
<td>142</td>
</tr>
<tr>
<td>9.2</td>
<td>Three Levels of Errors</td>
<td>143</td>
</tr>
<tr>
<td>9.3</td>
<td>Introducing the Exception</td>
<td>144</td>
</tr>
<tr>
<td>9.4</td>
<td>Using Exceptions</td>
<td>145</td>
</tr>
<tr>
<td>9.5</td>
<td>Tips: Test-Driven Development</td>
<td>146</td>
</tr>
<tr>
<td>9.6</td>
<td>Exercises</td>
<td>147</td>
</tr>
<tr>
<td>10</td>
<td>Developing Classes for Linear Algebra Calculations</td>
<td>151</td>
</tr>
<tr>
<td>10.1</td>
<td>Requirements of the Linear Algebra Classes</td>
<td>151</td>
</tr>
<tr>
<td>10.2</td>
<td>Constructors and Destructors</td>
<td>156</td>
</tr>
<tr>
<td>10.2.1</td>
<td>The Default Constructor</td>
<td>156</td>
</tr>
<tr>
<td>10.2.2</td>
<td>The Copy Constructor</td>
<td>156</td>
</tr>
<tr>
<td>10.2.3</td>
<td>A Specialised Constructor</td>
<td>157</td>
</tr>
<tr>
<td>10.2.4</td>
<td>Destructor</td>
<td>157</td>
</tr>
<tr>
<td>10.3</td>
<td>Accessing Private Class Members</td>
<td>157</td>
</tr>
<tr>
<td>10.3.1</td>
<td>Accessing the Size of a Vector</td>
<td>158</td>
</tr>
<tr>
<td>10.3.2</td>
<td>Overloading the Square Bracket Operator</td>
<td>158</td>
</tr>
<tr>
<td>10.3.3</td>
<td>Read-Only Access to Vector Entries</td>
<td>158</td>
</tr>
<tr>
<td>10.3.4</td>
<td>Overloading the Round Bracket Operator</td>
<td>158</td>
</tr>
<tr>
<td>10.4</td>
<td>Operator Overloading for Vector Operations</td>
<td>158</td>
</tr>
<tr>
<td>10.4.1</td>
<td>The Assignment Operator</td>
<td>159</td>
</tr>
<tr>
<td>10.4.2</td>
<td>Unary Operators</td>
<td>159</td>
</tr>
<tr>
<td>10.4.3</td>
<td>Binary Operators</td>
<td>159</td>
</tr>
<tr>
<td>10.5</td>
<td>Functions</td>
<td>159</td>
</tr>
<tr>
<td>10.5.1</td>
<td>Members Versus Friends</td>
<td>159</td>
</tr>
<tr>
<td>10.6</td>
<td>Tips: Memory Debugging Tools</td>
<td>160</td>
</tr>
<tr>
<td>10.7</td>
<td>Exercises</td>
<td>161</td>
</tr>
<tr>
<td>11</td>
<td>An Introduction to Parallel Programming Using MPI</td>
<td>165</td>
</tr>
<tr>
<td>11.1</td>
<td>Distributed Memory Architectures</td>
<td>165</td>
</tr>
<tr>
<td>11.2</td>
<td>Installing MPI</td>
<td>167</td>
</tr>
<tr>
<td>11.3</td>
<td>A First Program Using MPI</td>
<td>167</td>
</tr>
<tr>
<td>11.3.1</td>
<td>Essential MPI Functions</td>
<td>168</td>
</tr>
<tr>
<td>11.3.2</td>
<td>Compiling and Running MPI Code</td>
<td>169</td>
</tr>
<tr>
<td>11.4</td>
<td>Basic MPI Communication</td>
<td>171</td>
</tr>
<tr>
<td>11.4.1</td>
<td>Point-to-Point Communication</td>
<td>171</td>
</tr>
<tr>
<td>11.4.2</td>
<td>Collective Communication</td>
<td>174</td>
</tr>
</tbody>
</table>
11.5 Example MPI Applications
11.5.1 Summation of Series
11.5.2 Parallel Linear Algebra
11.6 Tips: Debugging a Parallel Program
11.6.1 Tip 1: Make an Abstract Program
11.6.2 Tip 2: Datatype Mismatch
11.6.3 Tip 3: Intermittent Deadlock
11.6.4 Tip 4: Almost Collective Communication
11.7 Exercises

12 Designing Object-Oriented Numerical Libraries
12.1 Developing the Library for Ordinary Differential Equations
12.1.1 Model Problems
12.1.2 Finite Difference Approximation to Derivatives
12.1.3 Application of Finite Difference Methods to Boundary Value Problems
12.1.4 Concluding Remarks on Boundary Value Problems in One Dimension
12.2 Designing a Library for Solving Boundary Value Problems
12.2.1 The Class SecondOrderOde
12.2.2 The Class BoundaryConditions
12.2.3 The Class FiniteDifferenceGrid
12.2.4 The Class BvpOde
12.2.5 Using the Class BvpOde
12.3 Extending the Library to Two Dimensions
12.3.1 Model Problem for Two Dimensions
12.3.2 Finite Difference Methods for Boundary Value Problems in Two Dimensions
12.3.3 Setting Up the Linear System for the Model Problem
12.3.4 Developing the Classes Required
12.4 Tips: Using Well-Written Libraries
12.5 Exercises

Appendix A Linear Algebra
A.1 Vectors and Matrices
A.1.1 Operations Between Vectors and Matrices
A.1.2 The Scalar Product of Two Vectors
A.1.3 The Determinant and the Inverse of a Matrix
A.1.4 Eigenvalues and Eigenvectors of a Matrix
A.1.5 Vector and Matrix Norms
A.2 Systems of Linear Equations
A.2.1 Gaussian Elimination
A.2.2 The Thomas Algorithm
A.2.3 The Conjugate Gradient Method
Appendix B  Other Programming Constructs You Might Meet . . . . 225
  B.1  C Style Output .................................................. 225
  B.2  C Style Dynamic Memory Allocation .............................. 226
  B.3  Ternary ?: Operator .............................................. 226
  B.4  Using Namespace .................................................. 227
  B.5  Structures .......................................................... 228
  B.6  Multiple Inheritance .............................................. 228
  B.7  Class Initialisers .................................................. 229

Appendix C  Solutions to Exercises ....................................... 231
  C.1  Matrix and Linear System Classes ............................. 231
  C.2  ODE Solver Library .............................................. 240

Further Reading .......................................................... 245
  Mathematical Methods and Linear Algebra ........................ 245
  C++ Programming ..................................................... 245
  The Message-Passing Interface (MPI) ................................ 245

Index ................................................................. 247