Guojun Gan
Department of Mathematics
University of Connecticut
Storrs, CT, USA

An Introduction to Excel VBA Programming: with Applications in Finance and Insurance

To all my students.

Contents

List of Figures xi

List of Tables xiii

Preface xv

I VBA Preliminaries 1

1 Introduction to VBA 3

1.1 Getting Started 4
1.2 Modules 9
1.3 VBA Classes 11
1.4 The Excel Macro Recorder 13
1.5 Summary 16

2 Excel Objects 17

2.1 The Excel Object Model 17
2.2 The Application Object 21
2.3 The Workbook Objects 23
2.4 The Worksheet Objects 28
2.5 The Range Object 32
2.6 The WorksheetFunction Object 37
2.7 Summary 41

3 Variables, Data Types, and Scopes 43

3.1 Variable Declaration and Data Types 43
3.2 Arrays 48
3.3 Constants 54
3.4 Strings 57
3.5 Dates 65
3.6 Scopes 69
3.7 Summary 73

4 Operators and Control Structures 75
4.1 Operators 75
4.2 Flow Control 81
4.3 Loops 86
4.4 Summary 93

5 Functions, Events and File IO 95
5.1 User-Defined Functions 95
5.2 Events 101
5.3 File IO 104
5.4 Summary 108

6 Error Handling and Debugging 109
6.1 Error Handling 109
6.2 Debugging VBA Code 115
6.3 Best Practices of VBA Coding 120
6.4 Summary 125

II Applications 127

7 Generating Payment Schedules 129
7.1 Introduction 130
7.2 Public Holidays in the United States 130
7.3 The Julian and Gregorian Calendars 131
7.4 Day Count Conventions 133
7.5 Business Day Conventions 134
7.6 Implementation 135
7.6.1 The MDate Module 135
7.6.2 The MHoliday Module 147
7.6.3 The MSchedule Module 151
7.6.4 The MInterface Module 156
7.7 Summary 159

8 Bootstrapping Yield Curves 161
8.1 Introduction 162
8.2 Interpolation 164
8.3 Bootstrapping Yield Curves 164
8.4 Finding Roots of an Equation 167
8.5 Implementation 168
8.5.1 The MCurve Module 168
8.5.2 The MInterface Module 179

8.6 Summary 181

9 Generating Risk-Neutral Scenarios 183

9.1 Introduction 184

9.2 The Black-Scholes Model 184

9.3 Generating Random Normal Numbers 186

9.4 Implementation 187

9.4.1 The MGenerator Module 187

9.4.2 The MInterface Module 195

9.5 Summary 197

10 Valuing a GMDB 199

10.1 Introduction 200

10.2 Life Table Construction 201

10.3 GMDB Valuation 202

10.4 Greek Calculation 205

10.5 Implementation 205

10.5.1 The MLifeTable Module 206

10.5.2 The MGMDB Module 209

10.5.3 The MInterface Module 214

10.6 Summary 216

11 Connecting to Databases 217

11.1 ActiveX Data Objects 218

11.2 SQL Queries 220

11.3 Implementation 221

11.3.1 The MDatabase Module 221

11.3.2 The MInterface Module 229

11.4 Summary 232

12 Object-Oriented Programming 233

12.1 Introduction 233

12.2 Objects 234

12.3 Implementation 234

12.3.1 The CZeroCurve Module 235

12.3.2 The CSwap Module 238

12.3.3 The CBootstrapper Module 241

12.3.4 An OO Bootstrapping Tool 246

12.4 Summary 249

A Solutions to Selected Exercises 251

A.1 Introduction to VBA 251
A.2 Excel Objects 251
A.3 Variables, Data Types, and Scopes 253
A.4 Operators and Control Structures 258
A.5 Functions, Events and File IO 260
A.6 Error Handling and Debugging 262
A.7 Generating Payment Schedules 262
A.8 Bootstrapping Yield Curves 266
A.9 Generating Risk-Neutral Scenarios 268
A.10 Valuing a GMDB 270
A.11 Connecting to Databases 273

References 275

Index 278

Index of VBA Keywords 281

List of Figures

1.1 The Excel 2013 interface in a Windows computer. 4
1.2 The Visual Basic Editor in a Windows computer. 5
1.3 The Excel 2010 interface in a Mac computer. 6
1.4 The Ribbon dialog in Excel 2010 in a Mac computer. 7
1.5 The Visual Basic Editor in a Mac computer. 8
1.6 The macro dialog. 8
1.7 A module displayed in the Visual Basic Editor. 9
1.8 Inserting a module in the Visual Basic Editor. 10
1.9 The Record Macro dialog. 14

2.1 The Excel object hierarchy. 18
2.2 The Object Browser in the Visual Basic Editor. 18
2.3 A screen shot of the U.S. Life Tables 1999-2001 - Males. 33
2.4 Worksheet functions in the Object Browser. 41

3.1 An illustration of explicit variable declaration. 48
3.2 VBA's built-in color constants. 56
3.3 VBA's built-in constants for variable types. 57

4.1 Conversion functions in the Visual Basic Editor. 77

5.1 Workbook events in the code window. 102

6.1 An example illustrating a run-time error. 110
6.2 An example illustrating another run-time error. 111
6.3 Checking values of VBA statements in the Immediate window in Break mode. 117
6.4 Setting a watch expression in the Watch window. 118
6.5 The Watches window and checking values of VBA statements in the Immediate window in Break mode. 118
6.6 The Locals window in Break mode. 119
6.7 Three tiers of an application and their relationships. 122

7.1 The interface of the payment schedule generator. 129
7.2 The interface of a holiday generator. 159

8.1 The interface of the yield curve bootstrapper. 161
8.2 A yield curve. 163

9.1 The interface of the risk-neutral scenario generator. 183
9.2 The first few rows and columns of the risk-neutral scenarios. 197

10.1 The interface of the GMDB valuation program. 199

11.1 The interface of the variable annuity inforce tool. 217
11.2 The inforce10k table in an Access database. 218
11.3 Checking the reference for Microsoft ActiveX Data Objects 2.8 Library. 219
11.4 The output in the worksheet when the first button is clicked. 231
11.5 The new table in the database when the second button is clicked. 231

List of Tables

2.1 Some commonly used properties of the `Application` object. 21
2.2 Some commonly used methods of the `Application` object. . 22
2.3 Some file extensions of Excel files. 25
2.4 Some commonly used properties of a `Workbook` object. . . . 25
2.5 Some commonly used methods of a `Workbook` object. 26
2.6 Two properties of the `Workbooks` object. 27
2.7 Some methods of the `Workbooks` object. 27
2.8 Some commonly used properties of a `Worksheet` object. . . 28
2.9 Some commonly used methods of a `Worksheet` object. . . . 30
2.10 Some methods of the `Worksheets` collection. 31
2.11 Various ways to refer to a `Range` object. 32
2.12 Some properties of a `Range` object. 35
2.13 Some methods of a `Range` object. 36
2.14 Some worksheet functions. 38

3.1 A list of built-in data types in VBA. 45
3.2 Some commonly used string functions in VBA. 59
3.3 VBA's built-in date and time functions. 66
3.4 Scopes of a variable. 70
3.5 Scopes of a constant. 72
3.6 Scopes of a procedure. 73

4.1 VBA's arithmetic operators. 75
4.2 VBA's string operator. 78
4.3 VBA's comparison operators. 79
4.4 VBA's logical operators. 80
4.5 VBA's programming constructs for flow control. 81
4.6 VBA's loops. 87

5.1 Some workbook events. 102
5.2 Some worksheet events. 103

6.1 Usage of the On Error statement. 113

6.2 Usage of the Resume statement. 113
6.3 Naming conventions. 120
6.4 Scope specifiers. 121
6.5 Array specifiers. 121
6.6 Data type specifiers. 121

7.1 Holidays and observances in the United States. 131
7.2 Parameters for calendar conversion algorithms. 132
7.3 Range names used by the payment schedule generator. 156
7.4 Dates of the Easter Sunday from 2010 to 2020. 160

8.1 Range names used by the yield curve bootstrapper. 162
8.2 Interest rates of various tenors. 162
8.3 US swap rates for various maturities on February 8, 2016. . . . 165

9.1 Range names defined in the interface of the risk-neutral scenario generator. 190

10.1 Range names used by the GMDB valuation program. 206

11.1 Range names used by the variable annuity inforce tool. . . . 217

Preface

Visual Basic for Applications (VBA) is a programming language created by Microsoft that can be used to automate operations in Microsoft Excel, which is perhaps the most frequently used computer software program for manipulating data and building models in banks and insurance companies. One advantage of VBA is that it enables you to do anything that you can do manually in Excel and do many things that Excel does not allow you to do manually. As a powerful tool, VBA has been used by actuaries and financial analysts to build actuarial and financial models.

In the spring of 2016, I was assigned to teach the course "Math3550: Programming for Actuaries," which is taken primarily by junior and senior undergraduate students majoring in actuarial science at the University of Connecticut. This course explores how an actuary uses computers to solve common actuarial problems and teaches students how to design, develop, test and implement programs using Microsoft Excel with VBA. Since existing books on Excel VBA do not have exercises or applications related to actuarial science, I started to write lecture notes for this course. This textbook has grown out of those lecture notes.

This textbook has been written for undergraduate students majoring in actuarial science who wish to learn the basic fundamentals and applications of Excel VBA. In doing so, this book does not assume that readers have any prior programming experience. This book will also be of use to actuaries and financial analysts working in insurance companies and banks who wish to learn Excel VBA.

This textbook is divided into two parts: preliminaries of Excel VBA programming and some applications of VBA in finance and insurance. The preliminaries covered in the first part include how to run VBA programs, modules, best practices of VBA coding, the Excel object model, variables, control statements, functions, and error handling, among many other things. The applications of VBA introduced in the second part include generating regular payment schedules, bootstrapping yield curves, creating risk-neutral scenarios, pricing a guarantee embedded in a variable annuity contract, how to connect to databases, and object-oriented programming in VBA.

The best way to learn programming is by doing. I encourage readers to practice the VBA code presented in the book. The book also contains many exercises. Sample solutions of some exercises are given in the appendix of this book. Readers should explore the exercises before looking at the solutions.

Finally, I would like to take this opportunity to express my thanks to my students, friends, and colleagues from the University of Connecticut who have read and provided valuable feedback on the draft of this book.

Guojun Gan
Storrs, Connecticut, USA
July 27, 2016