APPENDIX F

Accompanying CD ROM

All the code presented in this book is available in the accompanying CD ROM. The contents of the CD ROM is described in the sections of this chapter.

Permission to use the code for any purpose is granted to the purchaser of the book. The code is distributed on an "AS IS" basis, without warranty. Neither the author nor Morgan Kaufmann, Inc. make any representation, or warranty, either expressed or implied, with respect to the code, its quality or accuracy, or fitness for a specific purpose. Therefore, neither the author nor Morgan Kaufmann, Inc. shall have any liability to you or any other person or entity with respect to any liability, loss or damage caused or alleged to have been caused directly or indirectly by the code contained on the CD ROM. This includes, but is not limited to, interruption of service, loss of data, loss of classroom time, loss of consulting or anticipatory profits, or consequential damages from the use of the code contained on the CD ROM.

F.1 Smalltalk code

The Smalltalk code presented in this book was created with Visual Age for Smalltalk under ENVY. The entire code is contained in a single application. The application is split into several subapplications: these subapplications correspond more or less to the chapters of this book. To be honest, they correspond to the chapters of the early version.

F.1.1 non-ENVY users

Readers using Smalltalk without ENVY must load the classes in so-called *chunk* format. These files are located in the directory `Smalltlk` and all subdirectories. The names of the subdirectories and files correspond to the subapplications of the ENVY repository. These names have been modified to be used on systems using an 8.3 file name convention.

The classes of early chapters are referenced by classes of subsequent chapters. The reader should refer to figure 1.3 of chapter 1 for an explicit description of the dependency between classes.

Here is a description of the files:

`Smalltlk/DataMing/dataming.app` All classes defined in chapter 12.

`Smalltlk/Estim/genestim.app` All classes defined in chapter 10 and extensions of classes `DhbHistogram` and `DhbStatisticalMoments` related to the *t*-test.

`Smalltlk/Estim/estimChi.app` Extensions of classes `DhbHistogram` and `DhbStatisticalMoments` related to the χ^2-test.

`Smalltlk/Estim/estimF.app` Extensions of classes `DhbHistogram` and `DhbStatisticalMoments` related to the *F*-test.

`Smalltlk/Functs/general.app` The class `DhbPolynomial` defined in chapter 2 and all classes defined in chapter 3.

`Smalltlk/Functs/beta.app` Extensions of class `Number` defined in chapter 2 for the beta function.

Smalltlk/Functs/errfunct.app The class `DhbErfApproximation` defined in chapter 2 and extensions of class `Number` defined in chapter 2 for the error function.

`Smalltlk/Functs/gamma.app` The class `DhbLanczosFormula` defined in chapter 2 and extensions of classes `Number` and `Integer` defined in chapter 2 for the gamma function.

`Smalltlk/Iterativ/iterativ.app` All classes defined in chapters 4, 5, 6 and 7.

Smalltlk/LinAlgeb/linalgeb.app The classes defined in chapter 8, except for classes defined in the subapplications of this one.

`Smalltlk/LinAlgeb/eigenv.app` The classes `DhbLargestEigenValueFinder` and `DhbJacobiTransformation` defined in chapter 8.

`Smalltlk/LinAlgeb/determin.app` Extensions of classes `DhbMatrix` and `DhbLUPDecomposition` related to the computation of the matrix determinant.

`Smalltlk/LinAlgeb/inverse.app` Extensions of classes `DhbMatrix`, `DhbSymmetricMatrix` and `DhbLUPDecomposition` related to the computation of matrix inversion.

`Smalltlk/Minimize/minimize.app` All classes defined in chapter 11.

`Smalltlk/NumPrec/numprec.app` The class `DhbFloatingPointMachine` defined in chapter 1 and the class `DhbDecimalFloatingNumber` described in appendix A.

`Smalltlk/Statist/stats.app` All classes defined in chapter 9 as well as all classes describing a probability distribution (defined in chapters 9, 10 and appendix D).

F.1.2 ENVY users

Readers using ENVY™can load the ENVY™repository contained in the file `Vast/OONumS.dat`. This repository contains a single application, `DhbNumericalMethods`, containing the entire code presented in this book. Here is a brief description of the contents of each subapplication.

`DhbDataMining` All classes defined in chapter 12.

`DhbEstimation` All classes defined in chapter 10 and extensions of classes `DhbHistogram` and `DhbStatisticalMoments` related to the *t* test.

`DhbEstimationChi2` Extensions of classes `DhbHistogram` and `DhbStatisticalMoments` related to the χ^2 test.

`DhbEstimationF` Extensions of classes `DhbHistogram` and `DhbStatisticalMoments` related to the *F* test.

`DhbFunctionEvaluation` The class `DhbPolynomial` defined in chapter 2 and all classes defined in chapter 3.

`DhbBetaFunction` Extensions of class `Number` defined in chapter 2 for the beta function.

`DhbErrorFunction` The class `DhbErfApproximation` defined in chapter 2 and extensions of class `Number` defined in chapter 2 for the error function.

`DhbGammaFunction` The class `DhbLanczosFormula` defined in chapter 2 and extensions of classes `Number` and `Integer` defined in chapter 2 for the gamma function.

`DhbIterativeAlgorithms` All classes defined in chapters 4, 5, 6 and 7.

`DhbLinearAlgebra` The classes defined in chapter 8, except for classes defined in the subapplications of this one.

`DhbEigenValuesAndVectors` The classes `DhbLargestEigenValueFinder` and `DhbJacobiTransformation` defined in chapter 8.

`DhbMatrixDeterminant` Extensions of classes `DhbMatrix` and `DhbLUPDecomposition` related to the computation of the matrix determinant.

`DhbMatrixInversion` Extensions of classes `DhbMatrix`, `DhbSymmetricMatrix` and `DhbLUPDecomposition` related to the computation of matrix inversion.

`DhbMinimization` All classes defined in chapter 11.

`DhbNumericalPrecision` The class `DhbFloatingPointMachine` defined in chapter 1 and the class `DhbDecimalFloatingNumber` described in appendix A.

`DhbStatistics` All classes defined in chapter 9 as well as all classes describing a probability distribution (defined in chapters 9, 10 and appendix D).

F.2 Java code

The Java code presented in this book was created with Visual Age for Java. The entire code is split into several packages: these packages correspond more or less to the chapters of this book. They differ slightly from the Smalltalk applications because the Java code was generated independently from the Smalltalk code. Also due to the fact that Java does not allow to extend a class in another file, all files contain only one class.

The following packages have been defined:

`DhbDataMining` All classes defined in chapter 12.

`DhbEstimation` All classes defined in chapter 10.

`DhbFunctionEvaluation` All classes defined in chapters 1 and 2.

`DhbInterfaces` All interfaces defined in this book.

`DhbInterpolation` All classes defined in chapter 3.

`DhbIterations` All classes defined in chapters 4, 5, 6 and 7.

`DhbMatrixAlgebra` All classes defined in chapter 8.

`DhbOptimizing` All classes defined in chapter 11.

`DhbScientificCurves` The classes `Histogram` and `Curve`.

`DhbStatistics` The classes defined in chapter 9, except class `Histogram` and the classes defined in appendix D.

F.2.1 Visual Age for Java

The file `CDRom/Vaj/OONumJ.dat` contains a Visual Age repository. Users of Visual for Java can load the entire code by loading this file using the import facility of Visual Age.

F.2.2 Other Java systems

Classes must be loaded from the hierarchy of directories rooted at `CDRom/Java`. Each subdirectory corresponds to one of the packages described earlier.

note Because some Java system requires that the name of the class be identical to the name of the file, the original names of the classes have been kept. Similarly, the name of the directory must be equal to the name of the package. If you have transferred the contents of the CD ROM over a system limiting the size of the file name, you must first rename each file and directory before loading the classes.

F.3 Utilities

The directory `Utility` can only be used on a 32-bit Windows systems. It contains two programs to study some numerical effects experimentally. The directory must be copied as is onto a 32-bit Windows systems (Windows NT 4.0 or higher, Windows 98 or higher). The executable files must be run from this directory because it contains all the resources needed to execute them.

F.3.1 Distribution demo

The program contained in the file `DistributionDemo.exe` allows the reader to study all probability density functions described in this book. After starting the program, a window containing a note book will open. Each page of the notebook corresponds to a distribution. The parameters can be modified.

Then, clicking on the button labeled `Generate` will generate random values distributed according to the distribution. A window allows to modify the number of generated values and the parameters of the histogram into which the values are accumulated. When the generating is completed the resulting histogram is displayed. The reader can then attempt a least square fit or a maximum likelihood fit and view the result. The button labeled `Integral` allows to view the values as a distribution function.

F.3.2 File reader

This small utility program allows to display data read from a text file, either in tab or coma delimited format, in graphical form. Once data are read, the user can investigate interpolation, linear regression and polynomial least square fit. This program was used to generate the figures of chapters 3 and 10.

After starting the program, a window containing the parameters of the file containing the data will open. Here the reader can select the name of the file to read, the format used (blank or coma delimited) and which column contains which data. For error bars, there are three cases:

1. no error;
2. the error is given explicitly;
3. the error is computed from the standard deviation and the number of values used to compute the standard deviation (*c.f.* equation 9.6 in chapter 9).

If the data were read without error, the reader can investigate the three interpolation algorithms described in chapter 3. In addition linear regression and polynomial least square fit can be tried, each point having an identical weight.

If the data were read with errors, the reader can investigate only Lagrange and Bulirsch-Stoer interpolation. Linear regression and polynomial least square fit are performed using the supplied error to compute the weight of each point as described in chapter 10.