Understanding Ox code

Andrew P. Blake

1 What is Ox?

Ox is a matrix programming language developed by Jurgen Doornik. It is used to program PcGive and related software and can be set up to use through its own editor, very good, OxEdit.

This brief tutorial should help you read and create Ox code for many purposes. A comprehensive help can be found at Jurgen Doornik's website `http://www.nuff.ox.ac.uk/users/doornik/index.html`.

2 Sample program

In Table 2 we give a small example program that will illustrate almost all the basic principles. The output is in Table 2. Notice that the print statement in the program produces the formatted output shown in the table.

2.1 Basic points about Ox

Here are a few things to remember:

1. Ox is very like Gauss or Matlab. If you have used either you can use this.

2. All programs begin with the very 'C-like' `#include <oxstd.h>`. The `#include` is used to incorporate libraries — this is just the necessary standard one.

3. Statement terminate with a '`;`'. All subroutines programs and so on are contained within curly brackets, { }.

Table 1: Sample program

```
#include<oxstd.h>

// Routine that returns the square and the cube

squarecube(const i)
 {decl i2, i3;
  i2=i*i; i3=i^3;
  return{i2, i3}; }

// Main program

main()
 {decl x=5.75, y, z; format("%#12.5f");
  // Call routine
  [y, z] = squarecube(x);
  print("\nInput   = ", x, "\nSquared = ", y,
        "\nCubed   = ", z); }
```

4. Every variable needs to be declared using `decl`. Use can initialise as you declare, so `decl x=5.75;` both declares the variable and sets it to a number.

5. Procedure or subroutines are handy things to use, and can take arguments and have returns. The sample program shows the style.

6. Indexing is handled in a different way to many other matrix languages. Consider the matrix a such that:

$$a = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}$$

   Indexing in Ox starts at 0. So element a_{12} in Ox is `a[0][1]`. Similarly, element a_{31} in Ox is `a[2][0]`. This takes a bit of getting used to, but you soon do.

Table 2: Sample output

```
--------------- Ox at 10:41:08 on 27-Apr-2004 ---------------

Ox version 3.30 (Windows) (C) J.A. Doornik, 1994-2003

Input   =      5.75000
Squared =     33.06250
Cubed   =    190.10938
```

7. Matrix operations like concatenation and so on are often used, and algebra is very natural, with transposes, inverses and so on written quite naturally.

8. Use Jurgen Doornik's website for help!

Much of this is illustrated in the sample program. Ox programs look a little daunting at first because they look like 'proper' programs in a language like C. Look at the print statements. We mentioned the print statements above. In it:

```
"\n"
```

produces a new line, for example.

3 Why use Ox?

Ox comes can be used with many libraries written by econometricians such as James Davidson and David Hendry. These are freely available. The Console version of Ox is free for research and teaching purposes. The basic release allows access to estimating many models such as VARs and VECMs. Packages are also available for Long memory modelling, state-space (SSFpack), dynamic panel data (DPD), Markov-switching models (MS-VAR), Quantile regression as well as Lapack for linear algebra. This is very comprehensive. Whilst Ox is a little odd to begin with it is quick, flexible and the only language specifically designed with econometrics in mind.