AN INTRODUCTION TO OX

Sébastien Laurent

Universit de Namur and CORE, Louvain-la-Neuve

e-mail: Sebastien.Laurent@fundp.ac.be

http://www.core.ucl.ac.be/~laurent

Outline:

- 1) Introduction to the OxMetrics family
- 2) Installing Ox
- 3) Oxedit and GiveWin
- 4) Intoduction to Ox programming techniques
- 5) Maximization packages
- 6) Object-oriented programming
- 7) Graphics
- 8) OxGauss

OxMetrics

APPS	PcGive	STAMP	PcGets	TSP	Ox Packages
	+ *x12arima* + *PcNaive*				*DPD,MSVAR* *Arfima,G@RCH...*
CORE	GiveWin				Ox
	interactive graphics *data manipulation* *results storage* *code editor*				*numerical programming* *computational engine* *interface wrapper*

Why or why not Ox ?

Advantages

- It is free for academic use. This means you can download it and install it at home without breaking the law.
- It is fast and uses well-designed numeric routines that maintain accuracy.
- It is quite powerful. A good selection of built in mathematical functions, maximization routines, numeric integration and differentiation, etc., are included.
- It is extensible. You can write your own extensions, and there are downloadable add-on packages for estimating various models.
- The professional version is quite cheap.
- The graphical interface (GiveWin) of the Pro version is very nice.
- You can call Gauss procedures under Ox.

Disadvantages

- The free console version for Windows has poor graphics support. Graphics may be written to file, but not viewed on-screen without an external viewer such as Ghostscript. You can add on-screen graphics support using GnuDraw.

- Growing but smaller community than Gauss.

- The interactive mode is of little help (unlike Gauss).

First Step: Installing Ox

Contact

You can download a copy from

http://www.nuff.ox.ac.uk/Users/Doornik.

Or contact Timberlake Consultants. Timberlake can be found on the internet at `www.timberlake.co.uk` and `www.timberlake-consultancy.com`, or contacted via telephone in the UK on: +44 (0)20 8697 3377. Visit also *http://www.oxmetrics.net/*

Ox version

You can follow the course using Ox version 3 or newer (3.3), on any available computer platform.

2 versions:

(1) Console version (free for academics);

(2) Professional version.

Help and documentation

There is a reference book but much of that is also available in the online help. This help system is in the form of HTML documents, which can be read with an internet browser such as Netscape or the Internet Explorer. The help files can be found in the `ox\doc` directory; the master file is `index.html`. To read the file in the Internet Explorer, choose File/Open, then browse to find `index.html`, and then open it. The entries at the top give access to the *table of contents*, and to the *index*.

Running an Ox program

All versions of Ox which are free for educational purposes are *console versions*. This means that the program is launched from the command line in a console window (e.g. from the MS-DOS command prompt in a DOS window). Output will appear on the console as well. To run an Ox source code file called `myfirst.ox` in a console Window, issue the command (the `.ox` extension need not be typed):

```
oxl myfirst.ox
```

The output should be:

```
Ox version 3.30 (Windows) (C) J.A.Doornik, 1994-2003
two matrices
       2.0000      0.00000      0.00000
      0.00000       1.0000      0.00000
      0.00000      0.00000       1.0000

      0.00000      0.00000      0.00000
       1.0000       1.0000       1.0000
```

If the output is:

```
myfirst.ox (1): 'oxstd.h' include file not found
myfirst.ox (7): 'unit' undeclared identifier
myfirst.ox (11): 'print' undeclared identifier
```

then your include variable is not yet set.

If the output is something like 'bad command or filename', your path is not set. In Windows 95 and 98 the `PATH` variables are set by editing the autoexec.bat file.

```
Set PATH=...;C:\OX\Bin;
Set OX3PATH=C:\OX\include;C:\OX\;
```

In Windows NT/2000/XP, you can do it using the Control panel, System: use the environment page in the system properties.

In a moment we'll adopt more convenient ways to run Ox programs.

Redirecting output

Output from the console version appears on the console. To capture it in a file, *redirect* the output, e.g. to `myprog.out` as in:

```
oxl myfirst.ox > myprog.out
```

The `more` command may be used to page through large amounts of output (but you may prefer to use an editor): `oxl myfirst.ox | more`

Using the OxEdit editor

OxEdit is a powerful text editor, and a very useful program in its own right, see `www.oxedit.com`. Like GiveWin, OxEdit has some features which are especially useful when writing Ox programs:

- Syntax colouring.

  Three colours are used to distinguish keywords, constants and comment. This makes the code more readable, and mistakes easier to spot.

- Facility to easily comment in or comment out blocks of text.

- Run Ox programs from inside OxEdit.

- Context sensitive help.

  Just put the cursor on a word in the Ox source code, and press F1. For the index, use Help/Module Help Index.

- Double click on an error message to jump to the location of the error.

The first time you use OxEdit, execute the Add Predefined modules command View/Preferences menu, selecting Ox. From then on you can run your Ox programs without leaving OxEdit. The following commands are added to the Modules menu:

- Ox - runs the currently active document window using `oxl.exe`. The output will appear in the window called Ox Output.

  A shortcut for this is the 'running person' button on the toolbar.

- OxRun - runs the currently active document window using *OxRun*. The output will appear in `GiveWin`.

  A shortcut for this is the 'second running person' button on the toolbar.

- Ox - interactive - starts an interactive session. The input/output window is called Session.ox.

- Ox - debug - starts a debug session for the currently active document window. The input/output window is called Debug.ox.

You can even add more buttons representing Ox on the toolbar: right click on the toolbar (in the area next to a button), and add the relevant tool to the toolbar.

The Run icon is on the toolbar entitled 'Side bar', which is not shown by default. You can switch this on from the View menu.

Finally, OxEdit can highlight unbalanced parentheses, however, this is switched off by default. To activate got to Preferences/Options and check Show unbalanced parentheses.

Using GiveWin and OxRun

- GiveWin is part of Ox Professional, and offers some very useful features for developing and running Ox programs. The first thing to note when opening `myfirst.ox` using Open Text File on GiveWin's File menu is the syntax highlighting.

- There are several features to make programming easier. Unmatched parentheses are shown in red (forgetting a closing `)` or `}` is quite a common mistake). You can select of block of lines, and then use Comment In/Comment Out to make temporary changes (there is unlimited undo/redo as well):

- And very importantly, there is context sensitive help. For example, put the text cursor inside the word `print` in myfirst.ox. Then Press the F1 key.

- Select print() and press the Show Help button. This will start your Internet Explorer or Netscape at the point of the print function.

- To run a file, you can click on the Run button on the top toolbar.

- Finally, a mistake, for example omit the opening `(` parentheses after print, results in an error message. Just double click on the line with the error, and GiveWin will jump straight to the location of the error.

Bridging the gap between Ox and Gauss

Ox has the capability of running a wide range of Gauss programs. Gauss code can be called from Ox programs, or run on its own.

Running OxGauss programs from the command line

As an example we consider a small project, consisting of a code file that contains a simple procedure.

prod.prg

```
proc prod(x,y);
    local p;
    p=x*y;
    print "x*y=";;p;
    retp (p);
endp;

prod(3,4);
prod(5,6);
```

To run this program on the command line, enter

```
oxl -g prod.prg
```

To redirect the output in *prod.out*, you can use

```
oxl -g prod.prg > prod.out
```

Note that you can modify OxEdit in order to run gauss codes.

This produces:

```
---- Oxgauss at 11:15:29 on 31-Dec-2002 ----

Ox version 3.20j (Windows) (C) J.A. Doornik
x*y=       12.000000
x*y=       30.000000
```

Running OxGauss programs from GiveWin

Using Ox Professional, the OxGauss program can be loaded into GiveWin. The syntax highlighting makes understanding the program easier.

Click on Run (the running person) to execute the program. This runs the program using the OxGauss application, with the output in a window entitled OxGauss Session. GiveWin will treat the file as an OxGauss file if it has the .src, .g or .oxgauss extension.If not, the file can still be run by launching OxGauss from the GiveWin workspace window.

Calling Gauss procedure under Ox

There is a huge number of gauss procedures on the web. The main objective of creating OxGauss was to allow Gauss code to be called from Ox.

Here is a small example:

prod.src

```
proc prod(x,y);
    local p;
    p=x*y;
    print "x*y=";;p;
    retp (p);
endp;
```

prod.h

```
namespace gauss
{
    prod(const x, const y);
}
```

call_prod.ox

```
#include <oxstd.h>
#import "gauss::prod"

main()
{
    decl p;
    p=gauss::prod(10,3);
    print("p: ",p);
}
```

How does it work? When an OxGauss program is run, it automatically includes the *ox/include/oxgauss.ox* file. This itself imports the required files:

```
#define OX_GAUSS
#import <g2ox>
#import <gauss::oxgauss>
```

These import statements lead to g2ox.h and oxgauss.h being included. The majority of the OxGauss run-time system is in g2ox.ox. The keywords are largely in oxgauss.src.