WinSolve Version 4: DIY File Language

Richard Pierse

November 2007

This is a formal description of the algorithmic language used by WinSolve diy files. The first line of a diy file must start with the characters ‘WinSolve diy file’ (without the quotes). Commands and keywords are case insensitive and keywords can appear in any order. Each command should be on a single line unless otherwise stated. Lines starting with the ‘at’ character @ are treated as comments and ignored.

Conventions used in this description:

bold	typeface indicates a command or keyword that should be typed as shown
italic	typeface indicates a parameter or value to be supplied by the user
[ ]	denotes an optional keyword or parameter which may be omitted
|	denotes a set of (mutually exclusive) alternative keywords
. .	(ellipsis) denotes input is a series of values as indicated
//	indicates that the rest of the command should continue on a new line
bname	denotes a name to identify a block of equations
lname	denotes a name to identify a loop
vname	denotes the name of a variable in the current model

List of Commands

Looping Commands

loop *lname* [**alg** *alg*] [**itmax** *n*] [**itp** *n*] [**abs** *f*] [**pct** *f*]

Start an iteration loop with name *lname*.
Loop is terminated by matching ‘**end** *lname*’ command.

alg solution algorithm where *alg* is one of

gs	Gauss-Seidel (the default)
jacobi	Jacobi method
fgs	Fast Gauss-Seidel
newton	Newton method

itmax	maximum number of iterations
itp	iteration to commence printing unconverged variables
abs	absolute convergence criterion
pct	percentage convergence criterion

forw *lname*

Start a time loop with name *lname* running forwards in time.
Loop is terminated by matching ‘**end** *lname*’ command.

backw *lname*

Start a time loop with name *lname* running backwards in time.
Loop is terminated by matching ‘**end** *lname*’ command.

end *loopname*

End the loop *lname*. Loops must be nested correctly.

Solution Commands

term

Update terminal conditions.

solve *bname* [**damp** *f*]

Solve the block of equations *bname*.

damp solution damping factor to be used for equations in this loop

Equation Block Definitions

eqn *bname* // **[- | ~]***vname1* **. .** **[- | ~]***vnamek* **;**

Define block *bname* to consist of equations for variables: *vname1* to *vnamek*. These equations will be solved in the order that they are listed. List must be terminated by **;**

~ before a variable name indicates terminal condition equation
- before a variable name indicates *ignore* damping factor **damp** for this equation

An Example

The following is an example of a diy file that mimics the Fair-Taylor algorithm as implemented in WinSolve for a simple 12 equation model. The equations are split into 5 groups. *exogs* do not depend on any endogenous variables and can be solved before the outer iteration loop. *prerecs* are pre-recursive and so, in each time period, can be solved before the inner iteration loop. *simul* is the main simultaneous block. *postrecs* are post-recursive and so, in each time period can be solved after the inner iteration loop. Finally, *outputs* do not feed back into the other equations and so can be solved after the outer iteration loop. The equations in the *simul* block are solved with a damping factor of 0.7, except for variable Y8, where this damping is over-ridden.

WinSolve diy file

```
forw exog
        solve exogs
end exog
loop outer itmax 300 pct .0001
        term
        forw main
                solve prerecs
                loop inner
                        solve simul damp 0.7
                end inner
                solve postrecs
        end main
end outer
forw out
        solve outputs
end out
eqn exogs
        Y1 Y2 ;
```

```
eqn prerecs
        Y3 Y4 Y5 ;
eqn simul
        Y6 Y7 -Y8 Y9 ;
eqn postrecs
        Y10   ;
eqn outputs
        Y11 Y12 ;
```

[#] DIY
[k] DIY log file language
[#] DIYDEF
[k] DIY log file conventions
[#] DIYLOOPCOM
[k] Looping diy commands
[#] DIYLOOP
[k] loop diy command
[#] DIYFORW
[k] forw diy command
[#] DIYBACKW
[k] backw diy command
[#] DIYEND
[k] end diy command
[#] DIYSOLCOM
[k] Solution diy commands
[#] DIYTERM
[k] term diy command
[#] DIYSOLVE
[k] solve diy command
[#] DIYEQNCOM
[k] Equation block diy commands
[#] DIYEQN
[k] eqn diy command
[#] DIYEX
[k] Example of diy commands