

# EZ-SCROLL DOCUMENTATION

## I. General

This package uses OpenGL for 3D rendering. I routinely run EZ-Scroll on my PC with the Linux operating system. Note, you can run without graphics, but you must have OpenGL header files and libraries to use the code.

The EZ-Scroll package (in particular this document) is always under development. There are aspects of the code which I am not happy with, but to my knowledge everything works correctly.

The philosophy is to keep programs as simple as possible and to provide documentation by way of comments within the code itself. The user is expected to modify the programs according to his or her needs. The bulk of the package is devoted to graphics. Almost all of the execution time is spent in a loop in the routine `Step()` in *ezstep3d.c*.

The computational methods are described in more detail in the references at the end of this document. Ref. [5] describes the 3D implementation. Ref. [2] is the original source for the model. If you generate publications from using EZ-Scroll, I ask that you please cite these papers.

## II. Running EZ-Scroll

**Files:** You should have the following files:

*ezscroll.c*, *ezstep3d.c*, *ezgraph3d.c*, *ezmarching.c*, *ezscroll.h*, *ezstep3d.h*, *ezgraph3d.h*, *ezmarching.h*, *task.dat*, and *Makefile*.

You will probably want to save copies of these files (in compressed tar format).

**make:** It is up to you to edit *Makefile* as necessary for your system. I always use the GNU C compiler (gcc). You may, if you wish, specify NX etc. at compile time. Then these will be ignored in the task file. You can expect 10% to 20% improvement in speed if fix NX, NY, and NZ at compile time.

Make *ezscroll* by typing *make*. Then run by typing *ezscroll*. A window should open containing an initial condition for a scroll wave. The *u*-field is plotted. Hitting the space bar will start the simulation. This is a coarse resolution run showing the speed possible with EZ-Scroll simulations. With the pointer in the EZ-Scroll window, you can:

- (1) Switch between  $u$ -field,  $v$ -field, and no field by typing  $u$ ,  $v$ , or  $n$  respectively.
- (2) toggle filament plotting by typing  $f$ . (The filament is just the intersection of 2 contours.)
- (3) Toggle the clipping plane by typing  $c$ .
- (4) Pause the simulation by typing  $p$ , and resume by typing a space.
- (5) Rotate the image by first pausing the simulation, then by holding down the left mouse button and moving the cursor.
- (6) The key  $r$  resets the view to the initial (start up) view, and  $z$  sets the view to looking down the  $z$ -axis with the  $x$ - and  $y$ -axes in the usual orientation. This view is useful for moving the scroll.
- (7) The arrow keys move the scroll in the  $x$ - $y$  directions. The  $+$  key moves the scroll up the  $z$ -axis and the  $-$  moves it down the  $z$ -axis. Again, for moving the scroll it is best first to have set the view by typing  $z$ .
- (8) Stop the simulation by typing:  
 $q$  for soft termination with all files closed or  
 $ESC$  for immediate termination without writing final conditions (equivalent to typing control-C from the shell).

After a successful run, you will have a file *fc.dat* in your directory which contains the final conditions of the run. If you copy this file to *ic.dat*, then the next time you run *ezscroll*, this file will be read and used as an initial condition.

#### IV. Equations and Parameters

The model reaction-diffusion equations are [1,2]:

$$\frac{\partial u}{\partial t} = \nabla^2 u + \epsilon^{-1} u(1 - u)(u - u_{th}(v)), \quad \frac{\partial v}{\partial t} = D_v \nabla^2 v + g(u, v)$$

The method employed in EZ-Scroll is (essentially) independent of the choice of the functions  $u_{th}(v)$  and  $g(u, v)$  and the user is free to set these to whatever is desired. See *ezstep3d.h*.

In the simplest case

$$u_{th}(v) = \frac{v + b}{a}, \quad g(u, v) = u - v,$$

so that  $a, b$ , and  $\epsilon$  are parameters of the reaction kinetics.  $D_v$  is the ratio of diffusion coefficients ( $D_u \equiv 1$  by choice of length scales). In addition, there are lengths specifying the simulation volume:  $L_x, L_y$ , and  $L_z$ . Of these I choose to specify only  $L_x$  and let  $L_y$  and  $L_z$

be determined from the number of grid points (see below). Thus the “physical” parameters in the simulation are: **a**, **b**, **1/epsilon**, **Lx** and **Dv**.

The “numerical” parameters for the simulation are: **nx**, **ny**, **nz** = number of spatial grid points in each direction, **ts** = time step as fraction of the diffusion stability limit, and **delta** = small numerical parameter [1,2]. From **nx** and **Lx** the grid spacing is determined and from this and **ny**, **nz** the lengths **Ly**, **Lz** are determined.

The other parameters set in *task.dat* are:

Number of time steps to take

Time steps per plot. Also set the number of time steps per filament computation.

**write\_filament**. Flag for writing filament data

Time steps per write to history file

$(i, j, k)$  history point

initial field

initial condition type.

simulation and rotation resolutions

output type

verbose

These are more or less self-explanatory. If **write\_filament** is non-zero then the filament will be computed every **Time steps per plot** (whether or not there is any graphics) and the filament data will be written to a file (*filament.dat*). Note: each line of filament data consists of the time and a pair a points defining a line segment on the filament.

If the  $(i, j, k)$  point is in the domain, then a time series at the  $(i, j, k)$  grid point will be saved every **Time steps per write** assuming it is not zero.

I leave it to you to look at different initial condition types at the end of *ezscroll.c*.

If **simulation resolution**=1, then all the points in the simulation grid are used for the iso-surface and filament computation. If **simulation resolution**=2, then every other point in each direction is used, i.e. 1/8 as many points. This results in faster graphics. The same applies to the **rotation resolution** for setting the resolution during rotations only. Note: the **simulation**

`resolution` sets the resolution of filament computation even in the absence of any graphics, so for accurate filament data set `simulation resolution` to 1.

The other place to look for “parameters” is in the header files. The main compilation parameters are in *ezscroll.h*. Note that `SPLIT` is 1 in the distribution, but I think 0 is a better choice except at large time steps (See *ezstep3d.c*). Many of the macro definitions in the other header files can be replaced with variables.

## References

- [1] D. Barkley, M. Kness, and L.S. Tuckerman, Phys. Rev. A **42**, 2489 (1990).
- [2] D. Barkley, Physica **49D**, 61 (1991).
- [3] D. Barkley, Phys. Rev. Lett **68**, 2090 (1992).
- [4] D. Barkley, Phys. Rev. Lett. **72**, 164 (1994).
- [5] M. Dowle, R.M. Mantel, and D. Barkley, Int. J. Bif. Chaos **7**, 2529 (1997).

Please send comments to [D.Barkley@warwick.ac.uk](mailto:D.Barkley@warwick.ac.uk)