GUIDE TO FRACDIM

Download the files `fracdim.c`, `correl.c`, `Makefile`, `fracdim.m`, `lorenz.m` and `loaddat.m` to your 12.006 directory. I suggest you create a separate directory for these calculations so as not to overwrite any files. To do this simply type

```
% mkdir chaos
% cd chaos
```

and then save, compile, and analyze all files in your chaos directory. To compile the C codes, type

```
% make fracdim
```

at the Unix prompt. Please try this as soon as possible. In case of any trouble in making the target fracdim, please contact me as soon as possible.

If you are getting strange results, the first thing you should check is your disk space.

`fracdim.m` is a Matlab function that interfaces with `fracdim`, a program written in C (source code is in `fracdim.c`). This program enables you to compute the fractal dimension of a time series by imbedding it in a phase space of higher dimension. The algorithm is the one described on pages 150 to 154 of BPV's book, **O**rder Within Chaos. If the program is taking too long (it can be fairly slow to run if you have a lot of points) then you can try with fewer points (a few hundred can give a reasonable rough guess).

The default parameters for `fracdim` are shown below.

membed=3	The maximum embedding dimension.
logimax=7	The number of boxes that the spaces is divided into. 7 seems to work pretty well.
mindelay=1	The minimum spacing between points used to compute $C(r)$. It can be useful to set this to be > 1 if the data is finely sampled or a fractal dimension of 1 will result.

fracdim EXAMPLES

Generate the curves of $C(r)$ vs ϵ for the Lorenz attractor up to an embedding dimension of 5 and plot in log-log space so that the fractal dimension ν can be found (remember that $C(r) = Ar^{\nu}$):

```
>> tspan=[0 50];
>> [t,X] = ode45('lorenz', tspan, [0.1 0.1 0.1]);
>> membed = 5;
>> logimax = 7;
>> mindelay = 1;
>> fracdim(membed, logimax, mindelay, X(:,1));
>> loaddat
>>
```

In the above example, X is the time series used and loaddat loads the correlation data into Matlab and they are stored in the variables tseries_e2, tseries_e3, tseries_e4, etc, respectively for embedding dimensions 2, 3, 4, etc. Note that in order for loaddat.m to work, membed must be defined and must be updated everytime you intend to change the embedding dimension. tseries_e2 contains 2 columns of data: the first column contains r and the second column contains $C(r)$.

```
>> format long
>> tseries_e2
tseries_e2 =

   1.0e+06 *

   0.00000050000000   2.73847300000000
   0.00000025000000   1.69972100000000
   0.00000012500000   0.91483600000000
   0.00000006250000   0.44638000000000
   0.00000003125000   0.18656400000000
   0.00000001562500   0.06527900000000
   0.00000000781250   0.02015400000000
   0.00000000390625   0.00564800000000
   0.00000000195312   0.00148700000000
   0.00000000097656   0.00039700000000
   0.00000000048828   0.00012500000000
   0.00000000024414   0.00003700000000
   0.00000000012207   0.00001300000000
   0.00000000006104   0.00000600000000
>>
```

These can be plotted in loglog plot:

```
>> loglog(tseries_e2(:,1), tseries_e2(:,2), '-');
>> xlabel('r');
>> ylabel('C(r)');
```

Use whatever technique you find most effective to estimate the correlation dimension. Typing 'help polyfit' is a good place to start. Also note that the data is stored with large values of r first.