Matlab File List Supplement

This file lists the Matlab m-files that are made available as supplementary material for generating selected figures in the book *Chaos*, authored by A.C. Fowler and M.J. McGuinness, Springer, 2019. The annotated Matlab code for these figures will be made available online as an additional resource in the same location as this README file, alongside the book. The file descriptions here are very brief. Some of these files and figures are given more detailed descriptions in the appendix to the book.

Mark McGuinness, December 2019.

Figure 1.9

The cusp of the map associated with the Lorenz equations is sharpened in the m-file *fig1_9.m*, by shooting for the centre of the cusp. Two starting points are needed that straddle the cusp and hence can be iterated on using interval bisection. We use the (x, y, z) locations of successive maxima in z on an orbit. An orbit is followed until it completes two circuits of one unstable spiral point, before crossing to circle the other unstable spiral point. The sign of x is used as a simple proxy for which spiral point is being visited.

The first point chosen is the maximum in z (and its associated x and y values) on an orbit that completes one circuit of the same fixed point, and the second is the next maximum, on the same orbit, which has been chosen to then cross over to the other spiral point before reaching a local maximum in z there. These two points then provide initial conditions giving orbits that straddle the cusp, and a new starting point is chosen halfway between them. The orbit through this point is computed, and the next maximum of z is checked to see whether x is positive or negative there. The new pair of initial points with orbits that straddle the cusp is now used. The process is repeated, always halving the distance between the two straddling points, until they are very close to each other. Note that this is a process that corresponds to shooting for the stable manifold of origin.

Figure 2.1

Despite its simplicity, we include the m-file *fig2_1.m* that we used to create the cobweb plots in Fig. 2.1, for completeness.

Figure 2.2

The m-file *fig2_2.m* generates Fig. 2.2, the bifurcation or cascade diagram for the logistic map equation (2.3).

Figure 2.17

The m-file *fig2_17.m* generates Figure 2.17, the bifurcation or cascade diagram for the logistic equation in the form $x_{n+1} = 1 - \mu x_n^2$, which has superposed upon it the iterates of origin.

Figure 3.3

The m-files *fig3_3a.m* and *fig3_3b.m* generate Figure 3.3 by solving equation (3.9) numerically with $\omega = 1$ and $\mu = -0.05$ for the first plot, and $\mu = 0.09$ for the second plot.

Figure 4.2

This figure is generated by the m-file fig4_5.m — see later in this description, and in the appendix, the descriptions of Figure 4.4.

Figure 4.3

The m-file *fig4_3.m* generates both plots in Figure 4.3, the potential and the phase portrait for the nonlinear oscillator (4.1) .

Figure 4.4

The file *fig4_4.m* generates Figure 4.4, and if run on all 86 r-values takes about 20 minutes to run on a 2016 Macbook pro laptop. The *action* (period times amplitude) of unstable periodic orbits that exist around the two nonzero stable spiral points for the Lorenz equations is plotted against the parameter r. The r values range from the value where a homoclinic orbit exists for origin, to the value where the spiral point becomes unstable. At each value of r, the unstable periodic orbit is found by a shooting method: if successive maxima in the z variable are increasing, or if the orbit flips from circling one nonzero spiral point to circling the other, the initial point is outside the unstable periodic orbit; otherwise the initial point is inside.

Figure 4.5

The file *fig4_5.m* generates Figure 4.5, the unstable homoclinic orbit that exists near $\sigma = 10$, $b = 8/3$, and $r = 13.926$. It is just a version of the Figure 4.4 file with a fixed value of r. It also generates Figure 4.2.

Figure 4.7

The file *fig4_7.m* generates Figure 4.7, similarly to Figures 4.5 and 4.4.

Figure 4.16

The file *fig4_16.m* generates Figure 4.16, a plot of an orbit that is close to a homoclinic orbit of Shil'nikov type. As described in the Appendix to Chaos, first a value of c was found that gives a homoclinic orbit of simple type, the *principal* homoclinic orbit. The file *fig4_16_Find_c.m* was used to find the value $c = 10.31344913424363$ by iterating backwards in time. This value of c is used in *fig4_16.m* to compute an orbit going forwards in time, that starts near the fixed point P_2, and spirals away before making a close approach to

P_2 again. This orbit looks like the principal homoclinic orbit. See the appendix for more detailed comments on shooting for the homoclinic orbit.

Figure 5.3

The file *fig5_3.m* generates Figure 5.3, a Poincaré section through the solutions of a Hamiltonian system with the Hénon-Heiles potential.

Figure 5.4

The file *fig5_4.m* generates Figure 5.4, iterates of the standard map for $K = 0.85$.

Figure 5.7

The file *fig5_7.m* generates Figure 5.7, iterates of the standard map for $K = 0.757$, showing secondary resonance.

Figure 5.13

The file *fig5_13.m* generates Figure 5.13, a Poincaré section through solutions of a three-body problem with a constant value for the Jacobi integral and $\epsilon = 0.001$. The plot shows values of $(r, \dot{r})$ when $\psi = \pi/2$. It takes about a minute to run when the longest run end times are set to 5E06; if the full plot is desired the end times should be multiplied by 1000 instead of 10, in line 77 of the code, which takes about an hour to run.

Figure 5.14

The file *fig5_14.m* generates Figure 5.14, a Poincaré section through solutions of a three-body problem with a constant value for the Jacobi integral like in Figure 5.13, but with $\epsilon = 0.04$.

Figure 5.15

The file *fig5_15.m* generates Figure 5.15, a Poincaré section through solutions of a three-body problem with a constant value for the Jacobi integral like in Figure 5.13, but with $\epsilon = 0.1$.

Figure 5.16

The file *fig5_16.m* generates Figure 5.16, contours of constant effective potential V_E for the restricted three-body problem.

Figure 5.17

The file *fig5_17.m* generates Figure 5.17, a series of egg-shaped Poincaré sections in the Hénon-Heiles Hamiltonian system for increasing values of the energy.

Figure 5.18

The file *fig5_18.m* generates Figure 5.18, iterates of Henon's area-preserving map for $\alpha = 1.328$. Figure 5.19 is a magnification of this figure near the saddle.

Figures 5.20, 5.21

The file *fig5_20.m* generates Figure 5.20, iterates of the standard map, for $K = 0.5$. Changing K to one generates Figure 5.21.

Figure 6.1

The file *fig6_1a.m* generates Figure 6.1 (a). It shows the solution of the delay-recruitment equation $\epsilon\dot{x}(t) = -x(t) + 3.8x(t-1)(1-x(t-1))$ with initial values in the t-range $[-1, 0]$ linearly interpolated on ten random numbers in the range $[0.2, 1]$. It uses the delay-differential equation solver `dde23`, and takes half a day to reach the time range plotted when tolerances are set to `1.0E-08` with $\epsilon = 0.01$.

The file *fig6_1b.m* generates Figure 6.1 (b), the solution to an AR process.

The file *fig6_1c.m* generates Figure 6.1(c), the NZ TWI or the Trade Weighted Index, which is the currency value of the NZ dollar against a 17 currency basket. The horizontal axis is time in working days since 6 Jan 2014. This file requires that the data be present in the same folder, in a file called `TWI.txt`. This data file is provided with these m-files.

Figure 6.3

The file *fig6_3.m* generates the plots in Figure 6.3, a plot of the solution $x(t)$ to the Lorenz equations, and its power spectrum based on the periodogram, for $2T = 100$. It also illustrates what happens if you look at a higher resolution spectrum.

Figure 6.4

The file *fig6_4.m* generates the first plot in Figure 6.4, a power spectrum of the solution $x(t)$ to the Lorenz equations, based on the periodogram, for $2T = 800$. The second plot in Figure 6.4 may be generated by changing the value of `tend` in the file *fig6_4.m* to 6400.

Figure 6.6

The file *fig6_6.m* generates the singular value decomposition (svd) of a solution to the Lorenz equations, after some noise is added, and produces Figure 6.6, the first few singular values computed. It also produces figures showing the clean solution and the noisy signal analysed by svd.

Figure 6.7

The file *fig6_7.m* uses the singular values shown in Figure 6.6 to filter the white noise out of the noisy signal by the process described in the book and in the appendix. Figures output by this file include figure 6.7, the solution $x(t)$ to the Lorenz equation together with a lagged plot, in red; then the raw signal obtained by adding white noise to $x(t)$, then the filtered signal obtained by using just the first three singular values, then the result of filtering this filtered signal again by keeping only the first three singular values of its svd. Also shown is the result of filtering one more time, and the singular values that arise at each stage of the filtering process.

Figure 6.9

The file *fig6_9.m* generates the black spleenwort shown in figure 6.9. It uses the transformation matrices given in Michael Barnsley's book Fractals Everywhere, and the random iteration algorithm also to be found in that book.

Figure 6.10

The file *fig6_10.m* generates figure 6.10, the Julia set for the map $z^2 - \mu$, with complex z and $\mu = 0.999 - 0.251i$.

The Julia set is found by iterating the inverse of the function, starting with an unstable fixed point of the map, that is, $(1+\sqrt{1+4\mu})/2$. This approach is based on the observation that the Julia set is invariant under the map and under its inverse, and its inverse is not subject to rapid growth in magnitudes of errors. However, there are two inverses of any given point and hence as one iterates backwards the number of points increases exponentially fast, blowing out computer memory.

We follow the algorithm used in maxima code written by Adam Majewski (fraktal.republika.pl) with his permission. It is called the modified inverse iteration method (MIIM). This algorithm pushes the inverse iterates to a stack to store them, and limits the number of hits that are made on each pixel that is to be plotted, and pops from the stack to continue iterating. The stack grows rapidly at first, but eventually the stack is emptied and then iteration stops. Saved points are fully accurate to the default double precision for plotting and for iterating.

This process avoids the known issues with the doubling of the number of inverse points each iteration. More details are given in the code and in the appendix to our book.

Figure 6.11

The file *fig6_11.m* generates figure 6.11, the Julia set for the map $z^2 - \mu$, with complex z and a range of μ values. It uses the same method, MIIM, as described in Figure 6.10.

Figure 6.12

The file *fig6_12.m* generates the Mandelbrot set plotted in figure 6.12. It uses an escape time algorithm, together with a transformation that focusses the region of rapid colour changes more sharply onto the more interesting features of the set. The "hot" colourmap built into matlab is used, with 10,000 colours to reduce the banding that is seen when the escape time is integer valued. Points inside the basins of attraction that would normally be coloured white because they never escape, are set to black for better contrast at the boundary. The distances obtained are morphed to focus on the boundary by using two arctan functions, hand-tuned for a good-looking image. The resolution of the image that is used for this figure is 10,000 by 10,000 pixels. Since this takes about twelve hours to run on a Macbook Pro, we have also included Matlab Figure files (with names ending in .fig) for the high-resolution image and for its reverse colour-map version; the m-file is set to an image resolution of 1000 by 1000 pixels, so that it runs in a reasonably short time.

Mark McGuinness
Kilkee, Ireland

June 2018