Appendix A

Index of Python Programs

Readers can download the Python program files via GitHub:

https://github.com/springer-math/dynamical-systems-with-applications-using-python

These files will be kept up-to-date and extra files will be added in the forthcoming years.

A.1 IDLE Python Programs

These files include solutions to the Exercises listed in Chapter 1.

```
euclid_algorithm.py --- See Exercise 10.
F2C.py --- See Exercise 1(a).
F2K.py --- Converts degrees Fahrenheit to Kelvin.
fibonacci.py --- Lists first n terms of the Fibonacci sequence.
fmu.py --- The logistic function.
fractal_tree.py --- Plots a fractal tree.
fractal_tree_color.py --- Plots a color fractal tree.
grade.py --- Converts a score to a grade.
guess_number.py --- Guess the number game.
koch_snowflake.py --- See Exercise 1(d).
koch_square.py --- Plots a Koch square fractal.
Pythag_Triples.py --- See Exercise 1(c).
sierpinski.py --- Plots a Sierpinski triangle fractal.
sierpinski_square.py --- Plots a Sierpinski square fractal.
```


S. Lynch, *Dynamical Systems with Applications using Python*,
https://doi.org/10.1007/978-3-319-78145-7

```
sum_primes.py --- See Exercise 1(b).
sum_n.py --- Sums the natural numbers to n.
```

A.2 Anaconda Python Programs

If you have difficulty with the animation programs in Spyder, you have to change the backend to run an animation in the IPython console. You can do that by running

```
In[1]: %matplotlib qt5
```

before the animation. If you don't want to use this command every time, you can go to: Tools, Preferences, IPython Console, Graphics, Backend, and change it from "Inline" to "Automatic."

```
Program_01a.py --- Solve a simple ODE.
Program_01b.py --- Solve a second order ODE.
Program_01c.py --- Plot two curves on one graph.
Program_01d.py --- Subplots.
Program_01e.py --- Surface and contour plot in 3D.
Program_01f.py --- A parametric curve in 3D.
Program_01g.py --- Animation of a simple curve.

Program_02a.py --- Solve a separable ODE.
Program_02b.py --- Solve the logistic ODE.
Program_02c.py --- Power series solution.
Program_02d.py --- Power series solution for van der Pol.
Program_02e.py --- Plot series solution against numerical
                   solution.
Program_02f.py --- Solve a linear first order ODE.
Program_02g.py --- Solve a linear second order ODE.

Program_03a.py --- Plot the phase portrait of a linear system.
Program_03b.py --- Plot the phase portrait of a nonlinear
                   system.
Program_03c.py --- Finding critical points.

Program_04a.py --- Phase portrait and time series of Holling-
                   Tanner model.
```

```
Program_05a.py --- Limit cycle of a Fitzhugh-Nagumo system.
Program_05b.py --- Approximate and numerical solutions to ODEs.
Program_05c.py --- Error between one-term and numerical
                   solution.
Program_05d.py --- Lindstedt-Poincare technique.

Program_06a.py --- Contour plot.
Program_06b.py --- Surface plot.

Program_07a.py --- Animation of a simple curve.
Program_07b.py --- Animation of a subcritical Hopf bifurcation.
Program_07c.py --- Animation of a SNIC bifurcation.

Program_08a.py --- The Rossler attractor.
Program_08b.py --- The Lorenz Attractor.
Program_08c.py --- The Belousov-Zhabotinsky reaction.
Program_08d.py --- Animation of a Chua circuit bifurcation.

Program_09a.py --- Simple Poincare return map.
Program_09b.py --- Hamiltonian with two degrees of freedom plot.
Program_09c.py --- Phase portrait and Poincare map for the
                   Duffing system.
Program_09d.py --- Bifurcation diagram of Duffing equation.

Program_10a.py --- Computing Lyapunov quantities.
Program_10b.py --- Division algorithm for multivariate
                   polynomials.
Program_10c.py --- S-polynomial.
Program_10d.py --- Computing the Groebner basis.
Program_10e.py --- Computing Groebner basis of Lyapunov
                   quantities.
Program_10f.py --- Animation of a homoclinic limit cycle
                   bifurcation.
Program_10g.py --- Animation of a homoclinic limit cycle
                   bifurcation.

Program_11a.py --- Animation of a Lienard limit cycle.

Program_12a.py --- The method of steps.
Program_12b.py --- Plot of solution by method of steps.
Program_12c.py --- The Mackey-Glass DDE.
Program_12d.py --- The Lang-Kobayashi DDEs.
```

```
Program_13a.py --- Computing bank interest.
Program_13b.py --- Solving a second order recurrence relation.
Program_13c.py --- The Leslie matrix, eigenvalues and
                   eigenvectors.

Program_14a.py --- Graphical iteration of the tent map.
Program_14b.py --- Bifurcation diagram of the logistic map.
Program_14c.py --- Computing Lyapunov exponents for the
                   logistic map.
Program_14d.py --- Iteration of the Henon map.
Program_14e.py --- Lyapunov exponents of the Henon map.

Program_15a.py --- Point plot for a Julia set.
Program_15b.py --- Colormap of a Julia set.
Program_15c.py --- Color Mandelbrot set.
Program_15d.py --- Color Newton fractal Julia set.

Program_16a.py --- Intersection of implicit curves.
Program_16b.py --- Chaotic Attractor of the Ikeda map
Program_16c.py --- Bifurcation diagram of the Ikeda map.

Program_17a.py --- The Koch curve.
Program_17b.py --- Chaos game and the Sierpnski triangle.
Program_17c.py --- Barnsley's fern.
Program_17d.py --- Subplots of tau, D_q and f(alpha)
                   multifractal spectra.

Program_18a.py --- Generating a multifractal image
Program_18b.py --- Counting pixels in a color image.
Program_18c.py --- Image and statistical analysis on
                   microbes.png
Program_18d.py --- Fast Fourier transform of a noisy signal
Program_18e.py --- Iterative map and power spectra
Program_18f.py --- Fast Fourier transform of Lena image
Program_18g.py --- Edge detection in Lena image

Program_19a.py --- Chaos control in the logistic map.
Program_19b.py --- Chaos control in the Henon map.
Program_19c.py --- Chaos synchronization between two Lorenz
                   systems.
Program_19d.py --- Generalized synchronization.

Program_20a.py --- The generalized delta learning rule.
```

```
Program_20b.py --- The discrete Hopfield network.
Program_20c.py --- Iteration of a minimal chaotic neuromodule.
Program_20d.py --- Bifurcation diagram of neuromodule.

Program_21a.py --- The Hodgkin-Huxley equations.
Program_21b.py --- The Fitzhugh-Nagumo half-adder.
Program_21c.py --- Phase portrait Josephson junction limit
                   cycle.
Program_21d.py --- Animated Josephson junction limit cycle.
Program_21e.py --- Pinched hysteresis of a memristor.
```

Index

S. Lynch, *Dynamical Systems with Applications using Python*,
https://doi.org/10.1007/978-3-319-78145-7