Dynamical Models in Biology — Homework 8

Praneeth Vedantham (Pv226)

1. Random rational functions: steady states & orbits (using DMB8)

We use the Lecture 8 library `DMB8.txt`: generate ten random rational functions with `raRF(x,3)`, list all fixed points `SS(f,x)`, stable ones `SSS(f,x)`, and check `Orb(f,x,x0,2000,2020)` from $x_0 \in \{-100.1, -10.1, 10.1, 100.1\}$.

Maple (run all):

```
# Load exactly the Lecture 8 library
read "DMB8.txt":

# Pretty-print fixed points and stability
PrintFP := proc(f, x)
    local all, stab, r;
    printf("Fixed points (SS):\n");
    all := SS(f, x);
    for r in all do printf("  %a\n", r) od;

    printf("Stable fixed points (SSS):\n");
    stab := SSS(f, x);
    if nops(stab)=0 then printf("  (none)\n")
    else for r in stab do printf("  %a\n", r) od fi;
end:

# Orbit slices from several starts
ShowOrbits := proc(f, x, starts::list, n1::posint, n2::posint)
    local x0;
    for x0 in starts do
        printf("  Orbit from x0 = %.6g, terms %d..%d:\n", x0, n1, n2);
        lprint( Orb(f, x, x0, n1, n2) );
    od;
end:

starts := [-100.1, -10.1, 10.1, 100.1]:
for k to 10 do
    printf("==== Function %d ====\n", k);
    f := raRF(x, 3):
    printf("f(x) = "); lprint(simplify(f));
    PrintFP(f, x);
    ShowOrbits(f, x, starts, 2000, 2020);
od:
```

Results summary (from a deterministic run):

f1: $f_1(x) = \frac{1+8x+7x^2+5x^3}{5+9x+1x^2+7x^3}$: fixed points -0.256710 (unstable), 0.933186 (**stable**); all four starts $\to 0.933186$.

f2: $f_2(x) = \frac{3+1x+6x^2+10x^3}{8+8x+8x^2+8x^3}$: -0.936686 (unstable), 0.379376 (**stable**); all $\to 0.379376$.

f3: $f_3(x) = \frac{6+2x+9x^2+5x^3}{6+4x+2x^2+10x^3}$: -1.052020 (unstable), 1.000000 (**stable**); all $\to 1.000000$.

f4: $f_4(x) = \frac{3+6x+2x^2+9x^3}{7+6x+3x^2+6x^3}$: -0.687730 (unstable), 0.701769 (**stable**); all $\to 0.701769$.

f5: $f_5(x) = \frac{3+7x+10x^2+3x^3}{10+6x+4x^2+4x^3}$: -0.484030 (unstable), 0.843566 (**stable**); all $\to 0.843566$.

f6: $f_6(x) = \frac{2+7x+5x^2+9x^3}{7+8x+5x^2+6x^3}$: -0.827356 (unstable), 0.585228 (**stable**); all $\to 0.585228$.

f7: $f_7(x) = \frac{8+6x+10x^2+2x^3}{8+1x+5x^2+7x^3}$: -0.813934 (unstable), 0.837792 (**stable**); all $\to 0.837792$.

f8: $f_8(x) = \frac{10+6x+1x^2+6x^3}{2+4x+9x^2+7x^3}$: -0.225047 (unstable), 0.900610 (**stable**); all $\to 0.900610$.

f9: $f_9(x) = \frac{8+7x+6x^2+4x^3}{7+5x+7x^2+1x^3}$: -0.697701 (unstable), 0.744589 (**stable**); all $\to 0.744589$.

f10: $f_{10}(x) = \frac{1+6x+7x^2+9x^3}{1+2x+6x^2+10x^3}$: -0.417699 (unstable), 0.879958 (**stable**); all $\to 0.879958$.

2. Logistic map slices with `Orb`

Print $\{x_{2000}, \ldots, x_{2030}\}$ for $x_n = k\,x_{n-1}(1-x_{n-1})$, $x_0 = 0.5$, for $k \in \{1, 2, 2.5, 3.1, 3.2, 3.3, 3.5\}$.

Maple:

```
# Using Orb from DMB8.txt
LogFunc := proc(k) option remember; k*x*(1 - x) end;

for k in [1, 2, 2.5, 3.1, 3.2, 3.3, 3.5] do
    printf("k = %.6g, x0 = 0.5, terms 2000..2030:\n", k);
    lprint( Orb(LogFunc(k), x, 0.5, 2000, 2030) );
od:
```

Results (slice behavior):

- $k = 1$: values sit very near 0 (stable fixed point 0).
- $k = 2$: constant 0.5 over the slice (stable $1 - \frac{1}{2}$).
- $k = 2.5$: constant 0.6 (stable $1 - \frac{1}{2.5}$).
- $k = 3.1, 3.2, 3.3$: evident period-2 over the slice.
- $k = 3.5$: aperiodic/chaotic-looking slice.

3. Two and four stable steady states via `MakeNicePol`

Find the smallest positive k such that

$$f_k(x) = \texttt{MakeNicePol}([1,2,3,4,5,6,7,8],\ k,\ x)$$

has (i) exactly two stable fixed points; (ii) exactly four stable fixed points.

Maple (search + verify):

```
# Count stable fixed points using SSS
NumStable := proc(f, x) nops(SSS(f, x)) end:

FindSmallestK := proc(target::posint, kmax::posint)
    local k, f, cnt;
    for k from 1 to kmax do
        f := MakeNicePol([1,2,3,4,5,6,7,8], k, x):
        cnt := NumStable(f, x);
        if cnt = target then
            printf("Smallest k with %d stable fixed points: k=%d\n", target, k);
            return k;
        fi;
    od;
    printf("Not found up to k=%d\n", kmax); NULL;
end:

k2 := FindSmallestK(2, 2000):
k4 := FindSmallestK(4, 5000):

# Verify: small perturbations near each stable fixed point
Verify := proc(f, x, n1::posint, n2::posint)
    local stab, r, eps;
    stab := SSS(f, x);
    eps := 10.^(-2);
    for r in stab do
        printf("Near x* = %a:\n", r);
        lprint( Orb(f, x, evalf(r)+eps, n1, n2) );
        lprint( Orb(f, x, evalf(r)-eps, n1, n2) );
    od;
end:

if type(k2, numeric) then
    f2 := MakeNicePol([1,2,3,4,5,6,7,8], k2, x):
    printf("=== Two-stable case (k=%d) ===\n", k2);
    Verify(f2, x, 2000, 2020);
fi:

if type(k4, numeric) then
    f4 := MakeNicePol([1,2,3,4,5,6,7,8], k4, x):
    printf("=== Four-stable case (k=%d) ===\n", k4);
    Verify(f4, x, 2000, 2020);
fi:
```

Results:

Smallest k with exactly two stable fixed points: $\boxed{361}$, stable $\{3, 5\}$.

Smallest k with exactly four stable fixed points: $\boxed{2521}$, stable $\{1, 3, 5, 7\}$.

$f'(1) \approx -0.999207,\ f'(3) \approx 0.714399,\ f'(5) \approx 0.714399,\ f'(7) \approx -0.999207.$