Setup (Run these first!)

```
Off[General::spell];
Off[General::spell1];

ListOfNumbersQ[list_] := Module[{ret},
   If[ListQ[list] == False, ret = False,
    ret = True;
    Do[
     If[NumberQ[list[[i]]] == False, ret = False]
     , {i, 1, Length[list]}];
   ];
   Return[ret];
  ];

MatrixFunctionOftQ[mat_] := Module[{ret},
   If[MatrixQ[mat] == False, ret = False,
    ret = True;
    Do[
     If[ListOfNumbersQ[mat[[i]] /. t → Random[]] == False, ret = False]
     , {i, 1, Length[mat]}];
   ];
   Return[ret];
  ];

FloquetMultipliers[m_?MatrixFunctionOftQ, T_?NumberQ, opts___?OptionQ] :=
  Module[{x, sol, dim},
   dim = Length[m];
   sol = NDSolve[{x'[t] == m.x[t], x[0] == IdentityMatrix[dim]},
     x, {t, 0, T}, Flatten[{opts, Options[FloquetMultipliers]}]];
   Return[Sort[Eigenvalues[Evaluate[x[T] /. sol[[1]]]]]];
  ];
Options[FloquetMultipliers] = {MaxSteps → ∞};

FloquetExponents[m_?MatrixFunctionOftQ, T_?NumberQ, opts___?OptionQ] := Module[{},
   Return[Log[Chop[FloquetMultipliers[m, T, opts]]] / T];
  ];
Options[FloquetExponents] = {MaxSteps → ∞};

MaxFloquetMultiplier[m_?MatrixFunctionOftQ, T_?NumberQ] :=
  Module[{},
   Return[Max[Re[FloquetMultipliers[m, T]]]]];

MaxFloquetExponent[m_?MatrixFunctionOftQ, T_?NumberQ] :=
  Module[{},
   Return[Max[Re[FloquetExponents[m, T]]]]];

SmoothStep[p1_, p2_, p_, T_, k_] := Which[
   Mod[t, T] < 0.5 * p * T, p2 + (p1 - p2) * (0.5 + 0.31831 * ArcTan[k / T * Mod[t, T]]),
   Mod[t, T] < p * T + (1 - p) * 0.5 * T,
   p1 + (p2 - p1) * (0.5 + 0.31831 * ArcTan[k / T * (Mod[t, T] - p * T)]),
   True, p2 + (p1 - p2) * (0.5 + 0.31831 * ArcTan[k / T * (Mod[t, T] - T)])];
```

Example 1: Fitness of Structured Populations in Periodic

Environments

Setup

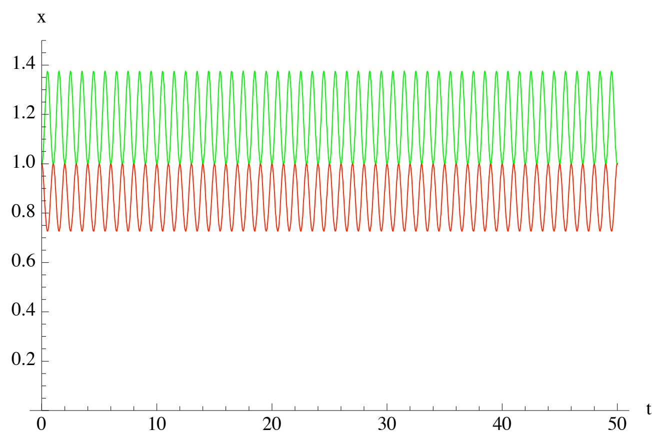
```
a[t] := {{Sin[2 π * t] - d, d}, {d, -Sin[2 π * t] - d}};
a[t] // MatrixForm
```

$$\begin{pmatrix} -d + \text{Sin}[2\,\pi\,t] & d \\ d & -d - \text{Sin}[2\,\pi\,t] \end{pmatrix}$$

Figure 1

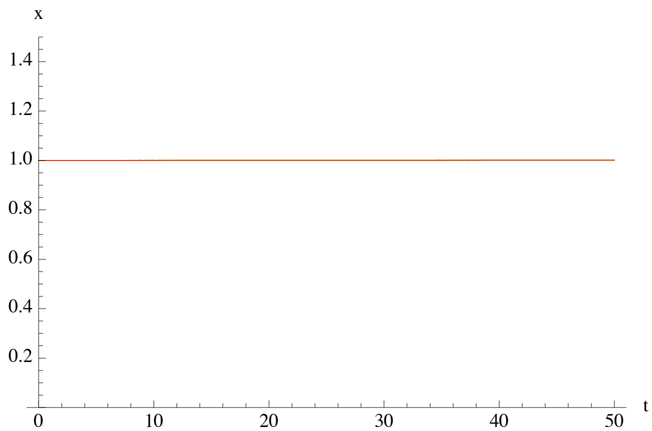
Part a, d=0

```
d = 0;
tmax = 50;
sol = NDSolve[{x′[t] == a[t].x[t], x[0] == {1, 1}}, x, {t, 0, tmax}];
Plot[{Evaluate[x[t] /. sol[[1]]][[1]], Evaluate[x[t] /. sol[[1]]][[2]]}, {t, 0, tmax},
 PlotStyle → {Green, Red}, AxesLabel → {"t", "x"}, PlotRange → {0, 1.5}]
```



▪ Part b, d=10^4 (effectively ∞)

```
d = 10^4;
tmax = 50;
sol = NDSolve[{x'[t] == a[t].x[t], x[0] == {1, 1}}, x, {t, 0, tmax}];
Plot[{Evaluate[x[t] /. sol[[1]]][[1]], Evaluate[x[t] /. sol[[1]]][[2]]}, {t, 0, tmax},
 PlotStyle → {Green, Red}, AxesLabel → {"t", "x"}, PlotRange → {0, 1.5}]
```



▪ Part c, d=3

```
d = 3;
tmax = 50;
sol = NDSolve[{x'[t] == a[t].x[t], x[0] == {1, 1}}, x, {t, 0, tmax}];
Plot[{Evaluate[x[t] /. sol[[1]]][[1]], Evaluate[x[t] /. sol[[1]]][[2]]}, {t, 0, tmax},
 PlotStyle → {Green, Red}, AxesLabel → {"t", "x"}, PlotRange → {0, 8}, PlotPoints → 100]
```

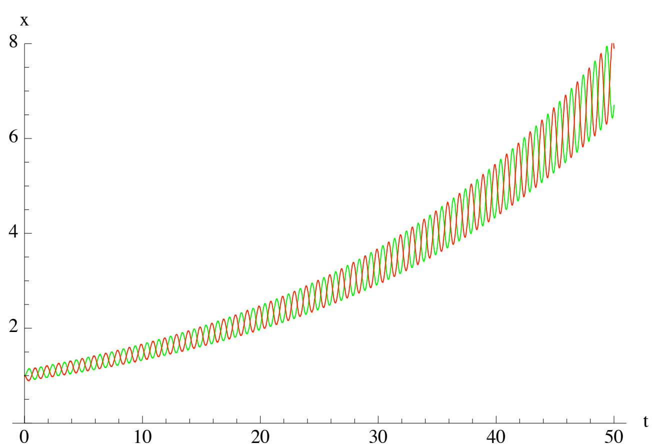
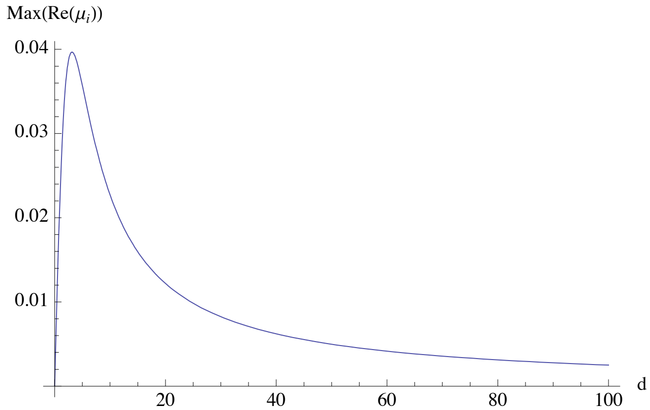


Figure 2

```
Clear[d];
```

```
Plot[MaxFloquetExponent[a[t], 1], {d, 0.001, 100},
 AxesLabel → {"d", "Max(Re(μi))"}, PlotRange → All]
```




```
FindMaximum[MaxFloquetExponent[a[t], 1], {d, 1, 2}]
```

```
{0.0396633, {d → 3.13173}}
```

Clean up

```
Clear[a, tmax, sol];
```

Example 2: Invasion Critera for Interacting Structured Populations

Setup

```
f1[r_] := bmax1 * r / (k1 + r);
f2[r_] := bmax2 * r / (k2 + r);

a = 1;
rin = 1;

y1 = 20;
m1 = 0.01;
d1j = 0.1;
d1a = 0.1;
bmax1 = 1;
k1 = 0.1;

y2 = 1;
m2 = 10;
d2j = 0.1;
d2a = 0.1;
bmax2 = 1;
k2 = 0.1;

k = 10^6; (* steepness parameter for smooth approximation to step function *)
```

Maximum growth rate and competitive ability

```
(* maximum growth rates in empty environment *)

λ10 = Max[Eigenvalues[{{-m1 - d1j, y1 * f1[rin]}, {m1, -d1a}}]]
λ20 = Max[Eigenvalues[{{-m2 - d2j, y2 * f2[rin]}, {m2, -d2a}}]]

0.321431

0.738742

(* competitive ability, R* *)

eq1 = Solve[{
    0 == a * (rin - r[t]) - n1a[t] * f1[r[t]],
    0 == y1 * f1[r[t]] * n1a[t] - m1 * n1j[t] - d1j * n1j[t],
    0 == m1 * n1j[t] - d1a * n1a[t]
   }, {r[t], n1j[t], n1a[t]}]

eq2 = Solve[{
    0 == a * (rin - r[t]) - n2a[t] * f2[r[t]],
    0 == y2 * f2[r[t]] * n2a[t] - m2 * n2j[t] - d2j * n2j[t],
    0 == m2 * n2j[t] - d2a * n2a[t]
   }, {r[t], n2j[t], n2a[t]}]
```

$\{\{n1j[t] \to -4.13407\times10^{-14}, n1a[t] \to -5.16758\times10^{-15}, r[t] \to 1.\},$
$\{n1j[t] \to 180.76, n1a[t] \to 18.076, r[t] \to 0.00582011\}\}$

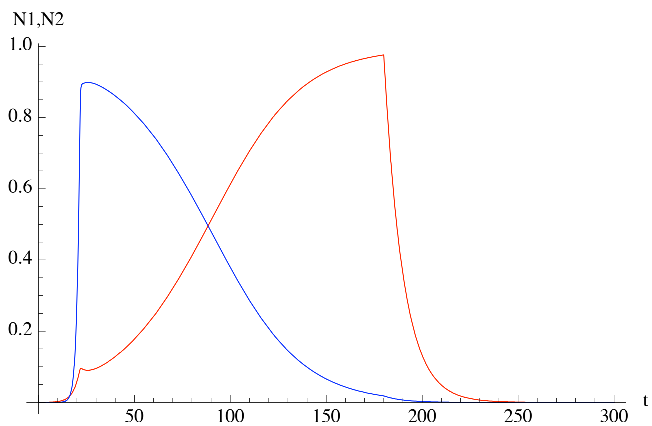
$\{\{n2j[t] \to -1.75877\times10^{-17}, n2a[t] \to -2.25122\times10^{-15}, r[t] \to 1.\},$
$\{n2j[t] \to 0.0978976, n2a[t] \to 9.78976, r[t] \to 0.0112347\}\}$

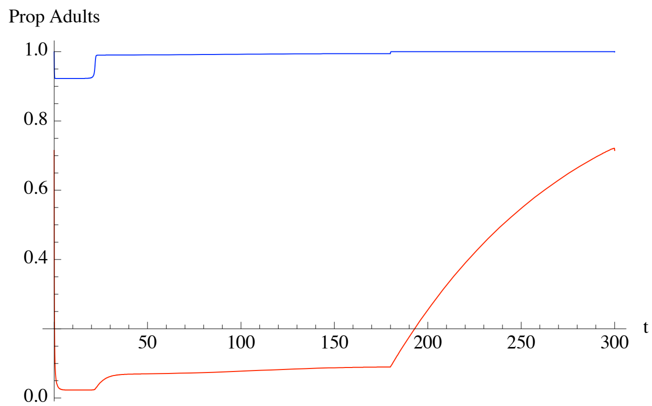
```
(* rate of invasion of monoculture equilibrium of species j by rare species 1 (λij) *)

λ21 = Max[Eigenvalues[{{-m2 - d2j, y2 * f2[r[t]]}, {m2, -d2a}}]] /. eq1[[2]]
λ12 = Max[Eigenvalues[{{-m1 - d1j, y1 * f1[r[t]]}, {m1, -d1a}}]] /. eq2[[2]]

-0.0452992

0.0372146

(* dying rate *)

λ1bad = -Max[d1j, d1a]
λ2bad = -Max[d2j, d2a]

-0.1

-0.1

(* solve for minimum p for species 1 and critical
 p for species 2 [slow fluctutaion approximation] *)

t1 = -λ1bad / λ10 * (1 - p);
Solve[t1 == p, p]
g2 = λ20 * t1 + λ21 * (p - t1) + λ2bad * (1 - p);
Solve[g2 == 0, p]

{{p → 0.237287}}

{{p → 0.760602}}
```

```
(* solve for minimum p for species 2 and critical
 p for species 1 [slow fluctutaion approximation] *)

t1 = -λ2bad / λ20 * (1 - p);
Solve[t1 == p, p]
g1 = λ10 * t1 + λ12 * (p - t1) + λ1bad * (1 - p);
Solve[g1 == 0, p]

{{p → 0.119226}}

{{p → 0.623111}}
```

Figure 3: View dynamics

```
(* Fig. 3A *)

T = 300;
p = 0.6;

tmax = T * 10;
ag = 512;

{ri, n1ji, n1ai, n2ji, n2ai} = {rin, 0.01, 0.01, 0.01, 0.01};

r[t] := rin - n1j[t] - n1a[t] - n2j[t] - n2a[t];

sol = NDSolve[{
    n1j'[t] == SmoothStep[y1 * f1[r[t]], 0, p, T, k] * n1a[t] - m1 * n1j[t] - d1j * n1j[t],
    n1a'[t] == m1 * n1j[t] - d1a * n1a[t],
    n2j'[t] == SmoothStep[y2 * f2[r[t]], 0, p, T, k] * n2a[t] - m2 * n2j[t] - d2j * n2j[t],
    n2a'[t] == m2 * n2j[t] - d2a * n2a[t],
    n1j[0] == n1ji, n1a[0] == n1ai, n2j[0] == n2ji, n2a[0] == n2ai},
   {n1j, n1a, n2j, n2a}, {t, 0, tmax}, MaxSteps → ∞, AccuracyGoal → ag];

fig3a1 =
 Plot[Evaluate[{n1j[t + tmax - T] + n1a[t + tmax - T], n2j[t + tmax - T] + n2a[t + tmax - T]} /. sol],
  {t, 0, T}, PlotStyle → {Red, Blue}, PlotRange → All, AxesLabel → {"t", "N1,N2"}]
fig3a2 = Plot[Evaluate[{n1a[t + tmax - T] / (n1j[t + tmax - T] + n1a[t + tmax - T]),
     n2a[t + tmax - T] / (n2j[t + tmax - T] + n2a[t + tmax - T])} /. sol], {t, 0, T},
  PlotStyle → {Red, Blue}, PlotRange → All, AxesLabel → {"t", "Prop Adults"}]

Clear[p];
```




Prop Adults
1.0
0.8
0.6
0.4
0.0
50
100
150
200
250
300
t

```
(* Fig. 3D *)

T = 10^4;
p = 0.68;

tmax = T * 10;
ag = 512;

{ri, n1ji, n1ai, n2ji, n2ai} = {rin, 0.01, 0.01, 0.01, 0.01};

r[t] := rin - n1j[t] - n1a[t] - n2j[t] - n2a[t];

sol = NDSolve[{
    n1j'[t] == SmoothStep[y1 * f1[r[t]], 0, p, T, k] * n1a[t] - m1 * n1j[t] - d1j * n1j[t],
    n1a'[t] == m1 * n1j[t] - d1a * n1a[t],
    n2j'[t] == SmoothStep[y2 * f2[r[t]], 0, p, T, k] * n2a[t] - m2 * n2j[t] - d2j * n2j[t],
    n2a'[t] == m2 * n2j[t] - d2a * n2a[t],
    n1j[0] == n1ji, n1a[0] == n1ai, n2j[0] == n2ji, n2a[0] == n2ai},
   {n1j, n1a, n2j, n2a}, {t, 0, tmax}, MaxSteps → ∞, AccuracyGoal → ag];

fig3b1 =
 Plot[Evaluate[{n1j[t + tmax - T] + n1a[t + tmax - T], n2j[t + tmax - T] + n2a[t + tmax - T]} /. sol],
  {t, 0, T}, PlotStyle → {Red, Blue}, PlotRange → All, AxesLabel → {"t", "N1,N2"}]
fig3b2 = Plot[Evaluate[{n1a[t + tmax - T] / (n1j[t + tmax - T] + n1a[t + tmax - T]),
     n2a[t + tmax - T] / (n2j[t + tmax - T] + n2a[t + tmax - T])} /. sol], {t, 0, T},
  PlotStyle → {Red, Blue}, PlotRange → All, AxesLabel → {"t", "Prop Adults"}]

Clear[p];
```

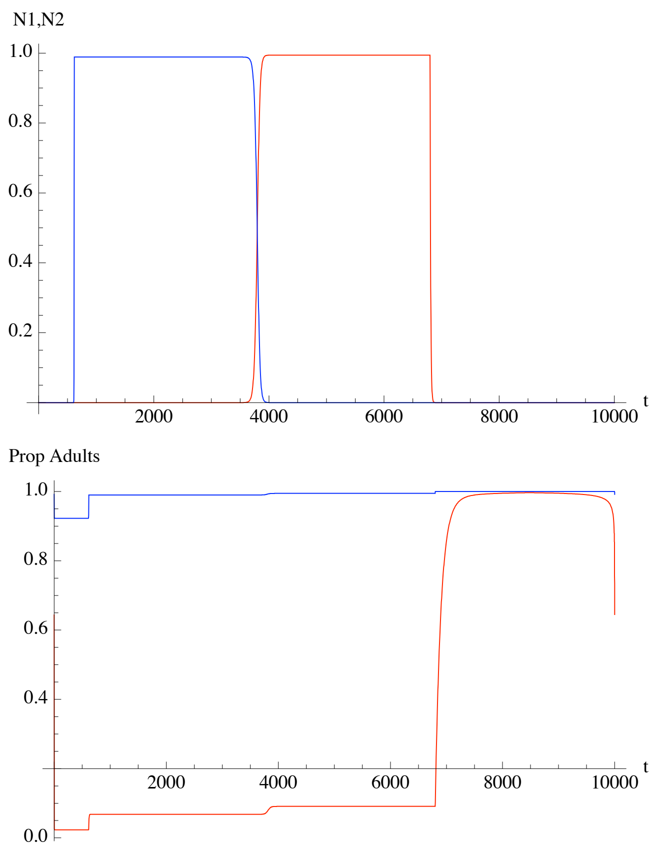


Figure 4: Outcome of competition

Setup various Floquet multipliers

```
(* floquet multiplier for invasion of species 1 into empty system *)

ρ10[p_?NumberQ] := Max[Re[Eigenvalues[MatrixExp[{{-m1 - d1j, 0}, {m1, -d1a}} * (1 - p) * T].
     MatrixExp[{{-m1 - d1j, y1 * f1[rin]}, {m1, -d1a}} * p * T]]]]

(* floquet multipliers for invasion of species 2 into empty system *)

ρ20[p_?NumberQ] := Max[Re[Eigenvalues[MatrixExp[{{-m2 - d2j, 0}, {m2, -d2a}} * (1 - p) * T].
     MatrixExp[{{-m2 - d2j, y2 * f2[rin]}, {m2, -d2a}} * p * T]]]]

(* floquet multipliers for invasion of monocultures *)

ρ21[p_?NumberQ] := Module[{tmp, n1ji, n1ai, sol1},

   {n1ji, n1ai} = {0.01, 0.01};

   sol1 = NDSolve[{
      n1j '[t] ==
       SmoothStep[y1 * f1[rin - n1j[t] - n1a[t]], 0, p, T, k] * n1a[t] - m1 * n1j[t] - d1j * n1j[t],
      n1a '[t] == m1 * n1j[t] - d1a * n1a[t],
      n1j[0] == n1ji, n1a[0] == n1ai},
     {n1j, n1a}, {t, 0, tmax}, MaxSteps → ∞, AccuracyGoal → ag];

   tmp = Re[Max[FloquetMultipliers[
        {{-m2 - d2j, SmoothStep[y2 * f2[rin - n1j[t + tmax - T] - n1a[t + tmax - T]], 0, p, T, k]},
          {m2, -d2a}} /. sol1[[1]], T, MaxSteps → ∞, AccuracyGoal → ag]]];

   Return[tmp];

  ];

ρ12[p_?NumberQ] := Module[{tmp, n2ji, n2ai, sol2},

   {n2ji, n2ai} = {0.01, 0.01};

   sol2 = NDSolve[{
      n2j '[t] ==
       SmoothStep[y2 * f2[rin - n2j[t] - n2a[t]], 0, p, T, k] * n2a[t] - m2 * n2j[t] - d2j * n2j[t],
      n2a '[t] == m2 * n2j[t] - d2a * n2a[t],
      n2j[0] == n2ji, n2a[0] == n2ai},
     {n2j, n2a}, {t, 0, tmax}, MaxSteps → ∞, AccuracyGoal → ag];

   tmp = Re[Max[FloquetMultipliers[
        {{-m1 - d1j, SmoothStep[y1 * f1[rin - n2j[t + tmax - T] - n2a[t + tmax - T]], 0, p, T, k]},
          {m1, -d1a}} /. sol2[[1]], T, MaxSteps → ∞, AccuracyGoal → ag]]];

   Return[tmp];
  ];

ag = 512; (* accuracy goal for NDSolve *)
agfr = 4; (* accuracy goal for find root *)
pgfr = 4; (* precision goal for find root *)
T := 10.0^Tpower;
```

When can each species invade the empty system? (two leftmost lines)

```
Tpowermin = -1;
Tpowermax = 4;
dTpower = 1 / 10;

(* solve for first point *)

Tpower = Tpowermin; pmin1new = pmin1i = pmin1[Tpower] =
   p /. FindRoot[Log[ρ10[p]] / T, {p, 0.12, 0.121}, AccuracyGoal → agfr, PrecisionGoal → pgfr];
pmin2new = pmin2i = pmin2[Tpower] = p /. FindRoot[Log[ρ20[p]] / T,
      {p, 0.2, 0.201}, AccuracyGoal → agfr, PrecisionGoal → pgfr];

Print[N[Tpowermin], " ", pmin1[Tpowermin], " ", pmin2[Tpowermin]];

(* solve for the rest (from T=10^-1 to 10^4, by 10^0.1) *)

Clear[Tpower];

Do[

  pmin1old = pmin1new;
  pmin2old = pmin2new;

  pmin1new = pmin1[Tpower] =
    p /. FindRoot[Log[ρ10[p]] / T, {p, pmin1i, 1.001 * pmin1i}, AccuracyGoal → agfr];

  pmin2new = pmin2[Tpower] =
    p /. FindRoot[Log[ρ20[p]] / T, {p, pmin2i, 1.001 * pmin2i}, AccuracyGoal → agfr];

  pmin1i = pmin1new + (pmin1new - pmin1old);
  pmin2i = pmin2new + (pmin2new - pmin2old);

  Print[N[Tpower], " ", pmin1[Tpower], " ", pmin2[Tpower]];

  , {Tpower, Tpowermin + dTpower, Tpowermax, dTpower}];
```

```
-1. 0.0604996 0.111172

-0.9 0.0604996 0.111214

-0.8 0.0604996 0.111277

-0.7 0.0604996 0.111373

-0.6 0.0604996 0.111515

-0.5 0.0604996 0.111719

-0.4 0.0604996 0.112001

-0.3 0.0604996 0.112369

-0.2 0.0604996 0.112824

-0.1 0.0605601 0.113358

0. 0.0605489 0.113954

0.1 0.0605982 0.114591

0.2 0.060623 0.115242

0.3 0.0607084 0.115878
```

```
0.4 0.0607938 0.116469
0.5 0.0609899 0.116992
0.6 0.0612769 0.117435
0.7 0.0617322 0.117799
0.8 0.0624551 0.118091
0.9 0.0636034 0.118325
1. 0.0654281 0.11851
1.1 0.0683247 0.118657
1.2 0.072897 0.118774
1.3 0.0799853 0.118867
1.4 0.0904687 0.118941
1.5 0.104528 0.119
1.6 0.12081 0.119046
1.7 0.137139 0.119083
1.8 0.152128 0.119113
1.9 0.165404 0.119136
2. 0.177015 0.119155
2.1 0.187092 0.119169
2.2 0.195773 0.119181
2.3 0.203196 0.11919
2.4 0.209489 0.119198
2.5 0.214776 0.119204
2.6 0.21917 0.119208
2.7 0.222783 0.119212
2.8 0.22572 0.119215
2.9 0.228083 0.119217
3. 0.229972 0.119219
3.1 0.231476 0.11922
3.2 0.232671 0.119222
3.3 0.23362 0.119223
3.4 0.234375 0.119223
3.5 0.234974 0.119224
3.6 0.235449 0.119224
3.7 0.235827 0.119225
3.8 0.236127 0.119225
```

```
3.9 0.236366 0.119225

4. 0.236555 0.119225
```

■ When can each species invade a monoculture of the other? (two rightmost lines)

```
(* note: this is slow *)

Tpowermin = 40 / 10;
Tpowermax = 17 / 10;
dTpower = -1 / 10;

tmax :=
  Which[Floor[Tpower] < 2.0, 40, Floor[Tpower] < 3.0, 20, Floor[Tpower] < 4.0, 10, True, 5] * T;

Tpower = Tpowermin; pcrit1new = pcrit1i = pcrit1[Tpower] = p /.
    FindRoot[Log[ρ12[p]] / T, {p, 0.6231, 0.62311}, AccuracyGoal → agfr, PrecisionGoal → pgfr];
pcrit2new = pcrit2i = pcrit2[Tpower] = p /. FindRoot[Log[ρ21[p]] / T,
      {p, 0.7606, 0.76061}, AccuracyGoal → agfr, PrecisionGoal → pgfr];
Print[N[Tpowermin], " ", pcrit1[Tpowermin], " ", pcrit2[Tpowermin]];

Clear[Tpower];
Do[
  pcrit1old = pcrit1new;
  pcrit2old = pcrit2new;
  pcrit1new = pcrit1[Tpower] = p /. FindRoot[Log[ρ12[p]] / T, {p, 0.999 * pcrit1i, 1.001 * pcrit1i},
      AccuracyGoal → agfr, PrecisionGoal → pgfr];
  pcrit2new = pcrit2[Tpower] = p /. FindRoot[Log[ρ21[p]] / T,
      {p, 0.999 * pcrit2i, 1.001 * pcrit2i}, AccuracyGoal → agfr, PrecisionGoal → pgfr];
  pcrit1i = pcrit1new + (pcrit1new - pcrit1old);
  pcrit2i = pcrit2new + (pcrit2new - pcrit2old);
  Print[N[Tpower], " ", pcrit1[Tpower], " ", pcrit2[Tpower]];
  , {Tpower, Tpowermin + dTpower, Tpowermax, dTpower}];
```

```
4. 0.619969 0.75647
3.9 0.619284 0.75555
3.8 0.618416 0.754383
3.7 0.617316 0.752907
3.6 0.615925 0.75104
3.5 0.614168 0.748683
3.4 0.611949 0.745712
3.3 0.60915 0.741981
3.2 0.605628 0.737322
3.1 0.601211 0.731548
3. 0.595712 0.72445
2.9 0.588925 0.715791
2.8 0.580631 0.705292
2.7 0.570589 0.692616
2.6 0.558529 0.677352
2.5 0.544138 0.658992
2.4 0.527045 0.636899
2.3 0.50681 0.610286
2.2 0.482905 0.578169
2.1 0.454696 0.539328
2. 0.421431 0.49223
1.9 0.382173 0.43491
1.8 0.335247 0.36472
1.7 0.275015 0.276736
```

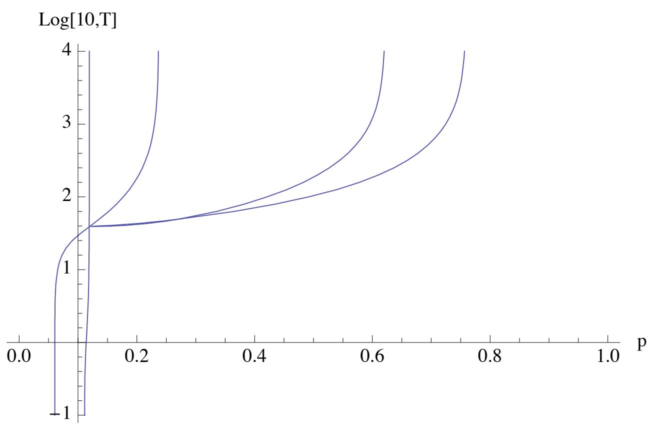
```
(* higher resolution from Tpower=1.60-1.70 *)

Tpowermin = 170 / 100;
Tpowermax = 160 / 100;
dTpower = -1 / 100;

Tpower = Tpowermin; pcrit1new = pcrit1i = pcrit1[Tpower] =
   p /. FindRoot[Log[ρ12[p]] / T, {p, 0.275, 0.2751}, AccuracyGoal → agfr, PrecisionGoal → pgfr];
pcrit2new = pcrit2i = pcrit2[Tpower] = p /. FindRoot[Log[ρ21[p]] / T,
      {p, 0.2767, 0.2768}, AccuracyGoal → agfr, PrecisionGoal → pgfr];
Print[N[Tpowermin], " ", pcrit1[Tpowermin], " ", pcrit2[Tpowermin]];

Clear[Tpower];
Do[
  pcrit1old = pcrit1new;
  pcrit2old = pcrit2new;
  pcrit1new = pcrit1[Tpower] = p /. FindRoot[Log[ρ12[p]] / T, {p, 0.999 * pcrit1i, 1.001 * pcrit1i},
      AccuracyGoal → agfr, PrecisionGoal → pgfr];
  pcrit2new = pcrit2[Tpower] = p /. FindRoot[Log[ρ21[p]] / T,
      {p, 0.999 * pcrit2i, 1.001 * pcrit2i}, AccuracyGoal → agfr, PrecisionGoal → pgfr];
  pcrit1i = pcrit1new + (pcrit1new - pcrit1old);
  pcrit2i = pcrit2new + (pcrit2new - pcrit2old);
  Print[N[Tpower], " ", pcrit1[Tpower], " ", pcrit2[Tpower]];
  , {Tpower, Tpowermin + dTpower, Tpowermax, dTpower}];
```

```
1.7 0.275 0.2767

1.69 0.267639 0.26651

1.68 0.259866 0.255911

1.67 0.251625 0.244882

1.66 0.24282 0.233344

1.65 0.233319 0.221179

1.64 0.222927 0.208205

1.63 0.211339 0.194102

1.62 0.198017 0.178212

1.61 0.181807 0.158603

1.6 0.158882 0.13278
```

```
(* even higher resolution from Tpower=1.593-1.60 *)

Tpowermin = 1600 / 1000;
Tpowermax = 1593 / 1000;
dTpower = -1 / 1000;

Tpower = Tpowermin;
pcrit1new = pcrit1i = pcrit1[Tpower] = p /. FindRoot[Log[ρ12[p]] / T, {p, 0.158881, 0.158883},
     AccuracyGoal → agfr, PrecisionGoal → pgfr];
pcrit2new = pcrit2i = pcrit2[Tpower] = p /. FindRoot[Log[ρ21[p]] / T,
      {p, 0.132779, 0.13281}, AccuracyGoal → agfr, PrecisionGoal → pgfr];
Print[N[Tpowermin], " ", pcrit1[Tpowermin], " ", pcrit2[Tpowermin]];

Clear[Tpower];
Do[
  pcrit1old = pcrit1new;
  pcrit2old = pcrit2new; pcrit1new = pcrit1[Tpower] = p /. FindRoot[Log[ρ12[p]] / T,
      {p, 0.9999 * pcrit1i, 1.0001 * pcrit1i}, AccuracyGoal → agfr, PrecisionGoal → pgfr];
  pcrit2new = pcrit2[Tpower] = p /. FindRoot[Log[ρ21[p]] / T,
      {p, 0.9999 * pcrit2i, 1.0001 * pcrit2i}, AccuracyGoal → agfr, PrecisionGoal → pgfr];
  pcrit1i = pcrit1new + (pcrit1new - pcrit1old);
  pcrit2i = pcrit2new + (pcrit2new - pcrit2old);
  Print[N[Tpower], " ", pcrit1[Tpower], " ", pcrit2[Tpower]];
  , {Tpower, Tpowermin + dTpower, Tpowermax, dTpower}];
```

```
1.6 0.158881 0.132779

1.599 0.155685 0.130886

1.598 0.152098 0.12919

1.597 0.147956 0.127659

1.596 0.142878 0.12626

1.595 0.136078 0.124944

1.594 0.128916 0.123563

1.593 0.124996 0.122271
```

```
plot1 = ListPlot[Table[{pmin1[Tpower], Tpower}, {Tpower, -1, 4, 1 / 10}], Joined → True];
plot2 = ListPlot[Table[{pmin2[Tpower], Tpower}, {Tpower, -1, 4, 1 / 10}], Joined → True];
plot3 = ListPlot[Table[{pcrit1[Tpower], Tpower}, {Tpower, 17 / 10, 4, 1 / 10}], Joined → True];
plot4 = ListPlot[Table[{pcrit2[Tpower], Tpower}, {Tpower, 17 / 10, 4, 1 / 10}], Joined → True];
plot5 = ListPlot[
   Table[{pcrit1[Tpower], Tpower}, {Tpower, 160 / 100, 170 / 100, 1 / 100}], Joined → True];
plot6 = ListPlot[Table[{pcrit2[Tpower], Tpower}, {Tpower, 160 / 100, 170 / 100, 1 / 100}],
   Joined → True];
plot7 = ListPlot[Table[{pcrit1[Tpower], Tpower}, {Tpower, 1593 / 1000, 1600 / 1000, 1 / 1000}],
   Joined → True];
plot8 = ListPlot[Table[{pcrit2[Tpower], Tpower}, {Tpower, 1593 / 1000, 1600 / 1000, 1 / 1000}],
   Joined → True];
Show[plot1, plot2, plot3, plot4, plot5, plot6, plot7, plot8,
 PlotRange → {{0, 1}, {-1, 4}}, AxesLabel → {"p", "Log[10,T]"}]
```



Clean up

```
Clear[plot1, plot2, plot3, plot4, plot5, plot6, plot7, plot8, pcrit1new,
 pcrit2new, pcrit1i, pcrit2i, Tpower, Tpowermax, Tpowermin, dTpower, T, tmax, ρ10,
 ρ20, ρ21, ρ12sol, r, ag, agfr, pgfr, λ10, λ20, λ21, λ12, λ1bad, λ2bad, eq1, eq2,
 a, rin, y1, m1, d1j, d1a, bmax1, k1, y2, m2, d2j, d2a, bmax2, k2, k, t1, g1, g2]
```

Ex 3: Stability of a Limit Cycle

Setup

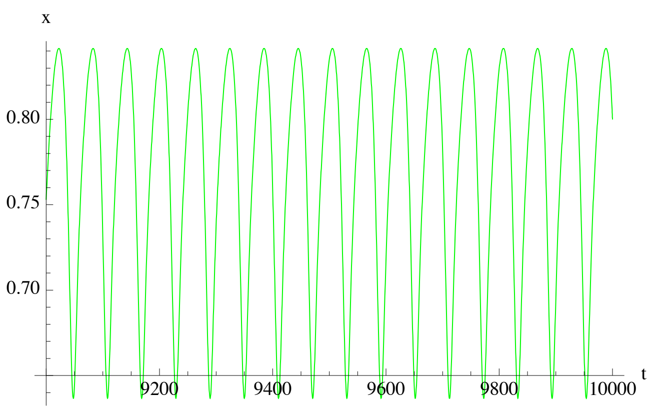
```
(* right hand sides *)

f1 := x[t] * (1 - x[t]) - a1 * x[t] / (1 + b1 * x[t]) * y[t];
f2 := a1 * x[t] / (1 + b1 * x[t]) * y[t] - a2 * y[t] / (1 + b2 * y[t]) * z[t] - d1 * y[t];
f3 := a2 * y[t] / (1 + b2 * y[t]) * z[t] - d2 * z[t];

(* jacobian matrix *)

j := D[{f1, f2, f3}, {{x[t], y[t], z[t]}, 1}];
```

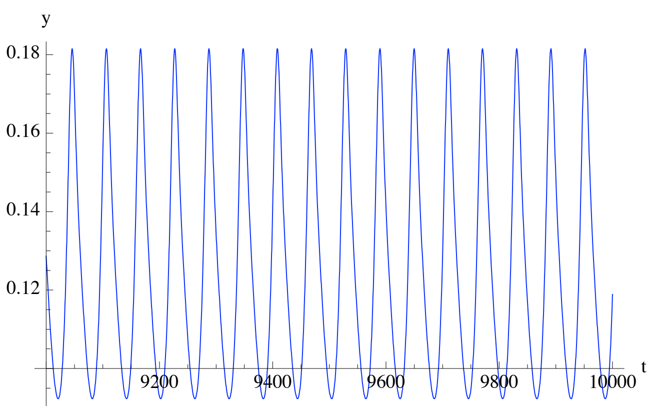
```
(* parameters *)

a1 = 5.0;
a2 = 0.1;
b2 = 2.0;
d1 = 0.4;
d2 = 0.01;

tmax = 10000;
```

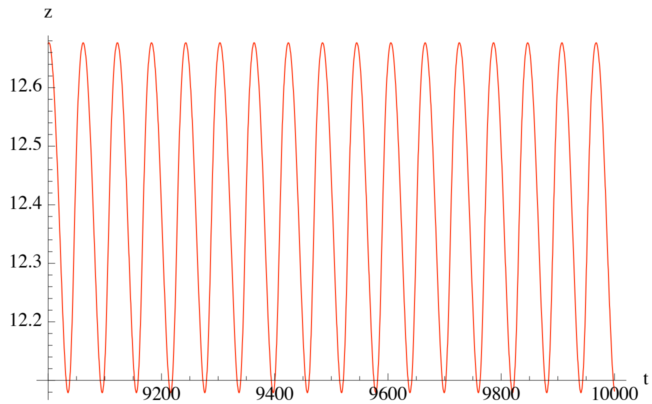
Figure 5

■ Part a

```
b1 = 2.28;

sol = NDSolve[{
    x'[t] == f1,
    y'[t] == f2,
    z'[t] == f3,
    x[0] == 0.1, y[0] == 0.1, z[0] == 5.0}, {x, y, z}, {t, 0, tmax}, MaxSteps → ∞];

Plot[Evaluate[x[t] /. sol[[1]]],
 {t, tmax - 1000, tmax}, PlotStyle → Green, AxesLabel → {"t", "x"}]
Plot[Evaluate[y[t] /. sol[[1]]], {t, tmax - 1000, tmax}, PlotStyle → Blue, AxesLabel → {"t", "y"}]
Plot[Evaluate[z[t] /. sol[[1]]], {t, tmax - 1000, tmax}, PlotStyle → Red, AxesLabel → {"t", "z"}]
```

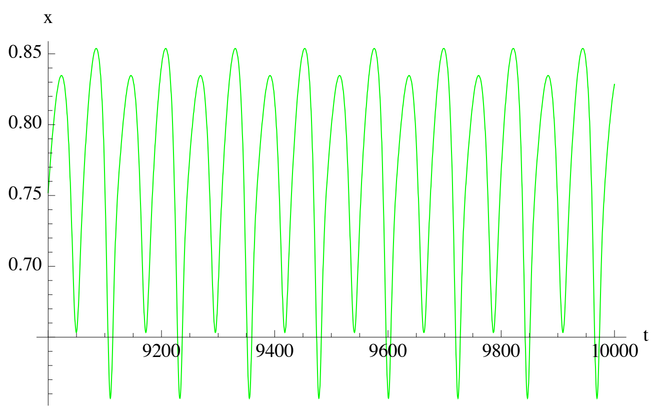


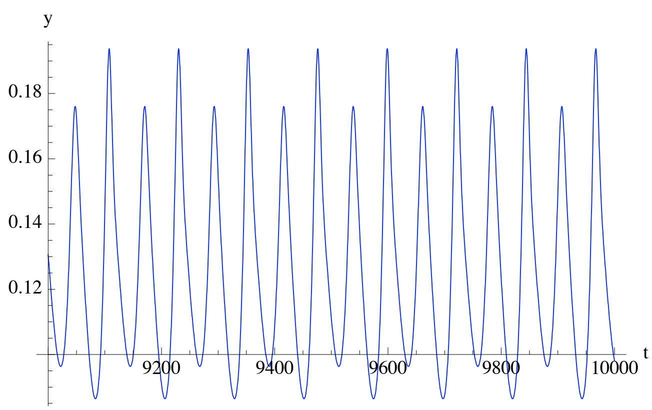


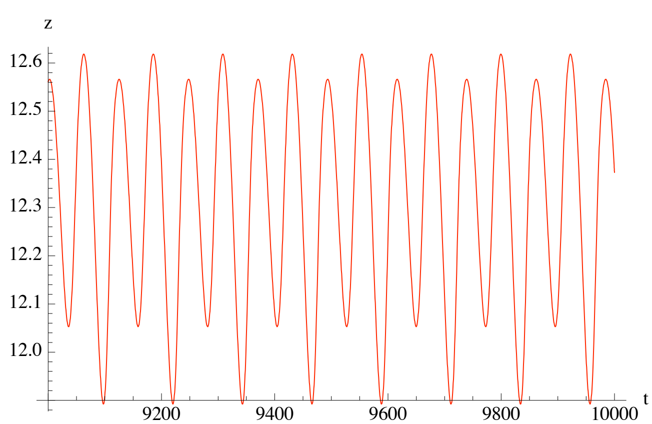

z
12.6
12.5
12.4
12.3
12.2
9200
9400
9600
9800
10000
t

■ Part b

```
b1 = 2.30;

sol = NDSolve[{
    x'[t] == f1,
    y'[t] == f2,
    z'[t] == f3,
    x[0] == 0.1, y[0] == 0.1, z[0] == 5.0}, {x, y, z}, {t, 0, tmax}, MaxSteps → ∞];

Plot[Evaluate[x[t] /. sol[[1]]],
 {t, tmax - 1000, tmax}, PlotStyle → Green, AxesLabel → {"t", "x"}]
Plot[Evaluate[y[t] /. sol[[1]]], {t, tmax - 1000, tmax}, PlotStyle → Blue, AxesLabel → {"t", "y"}]
Plot[Evaluate[z[t] /. sol[[1]]], {t, tmax - 1000, tmax}, PlotStyle → Red, AxesLabel → {"t", "z"}]
```







Part c

```
(* trial solution *)

trial[x0_?NumberQ, y0_?NumberQ, z0_?NumberQ, T_?NumberQ] := Module[{sol},

   sol = NDSolve[{
      x'[t] == f1,
      y'[t] == f2,
      z'[t] == f3,
      x[0] == x0, y[0] == y0, z[0] == z0}, {x, y, z}, {t, 0, T}, MaxSteps → ∞];

   Return[Evaluate[{x[T], y[T], z[T]} /. sol[[1]]]];
  ];

(* set initial conditions for loop *)

b1 = 2.27;
{xf, yf, zf} = trial[0.1, 0.1, 5.0, 10000];
Tf = 60;

Clear[b1];
Do[

  (* find limit cycle *)

  lc = FindRoot[{
     trial[x0, y0, z0, T] == {x0, y0, z0},
     z0 == zf},
    {x0, xf}, {y0, yf}, {z0, zf}, {T, Tf}];

  {xf, yf, zf, Tf} = Evaluate[{x0, y0, z0, T} /. lc];

  sol2 = NDSolve[
    {x'[t] == f1, y'[t] == f2, z'[t] == f3, x[0] == xf, y[0] == yf, z[0] == zf}, {x, y, z}, {t, 0, Tf}];

  (* calculate Floquet multipliers *)

  fm[b1] = Min[Re[FloquetMultipliers[j /. sol2[[1]], Tf]]];

  Print[b1, " ", lc, " ", fm[b1]];

  , {b1, 2.27, 2.31, 0.001}];
```

```
2.27 {x0 → 0.83509, y0 → 0.0970632, z0 → 12.2494, T → 60.0514} -0.807031
2.271 {x0 → 0.835802, y0 → 0.0966315, z0 → 12.2494, T → 60.0852} -0.816596
2.272 {x0 → 0.836471, y0 → 0.0962239, z0 → 12.2494, T → 60.119} -0.826132
2.273 {x0 → 0.837102, y0 → 0.0958389, z0 → 12.2494, T → 60.1529} -0.835637
2.274 {x0 → 0.837695, y0 → 0.0954749, z0 → 12.2494, T → 60.1869} -0.845107
2.275 {x0 → 0.838253, y0 → 0.0951308, z0 → 12.2494, T → 60.2209} -0.854545
2.276 {x0 → 0.838779, y0 → 0.0948053, z0 → 12.2494, T → 60.2549} -0.863951
2.277 {x0 → 0.839273, y0 → 0.0944976, z0 → 12.2494, T → 60.289} -0.873325
2.278 {x0 → 0.839738, y0 → 0.0942066, z0 → 12.2494, T → 60.3231} -0.882666
2.279 {x0 → 0.840175, y0 → 0.0939316, z0 → 12.2494, T → 60.3573} -0.891975
2.28 {x0 → 0.840586, y0 → 0.0936717, z0 → 12.2494, T → 60.3915} -0.901249
2.281 {x0 → 0.840971, y0 → 0.0934263, z0 → 12.2494, T → 60.4258} -0.910494
```

FindRoot::lstol :
The line search decreased the step size to within tolerance specified by AccuracyGoal and PrecisionGoal but was unable to find a sufficient decrease in the merit function. You may need more than MachinePrecision digits of working precision to meet these tolerances. ≫

```
2.282 {x0 → 0.841332, y0 → 0.0931947, z0 → 12.2494, T → 60.4601} -0.919699
```

FindRoot::lstol :
The line search decreased the step size to within tolerance specified by AccuracyGoal and PrecisionGoal but was unable to find a sufficient decrease in the merit function. You may need more than MachinePrecision digits of working precision to meet these tolerances. ≫

```
2.283 {x0 → 0.841673, y0 → 0.092975, z0 → 12.2494, T → 60.4945} -0.928972
2.284 {x0 → 0.841986, y0 → 0.0927708, z0 → 12.2494, T → 60.5289} -0.938014
2.285 {x0 → 0.84228, y0 → 0.0925773, z0 → 12.2494, T → 60.5634} -0.947121
2.286 {x0 → 0.842554, y0 → 0.0923957, z0 → 12.2494, T → 60.5979} -0.956194
2.287 {x0 → 0.842808, y0 → 0.0922254, z0 → 12.2494, T → 60.6325} -0.965232
2.288 {x0 → 0.843042, y0 → 0.0920661, z0 → 12.2494, T → 60.6671} -0.974236
2.289 {x0 → 0.843258, y0 → 0.0919174, z0 → 12.2494, T → 60.7018} -0.983205
```

FindRoot::lstol :
The line search decreased the step size to within tolerance specified by AccuracyGoal and PrecisionGoal but was unable to find a sufficient decrease in the merit function. You may need more than MachinePrecision digits of working precision to meet these tolerances. ≫

General::stop : Further output of FindRoot::lstol will be suppressed during this calculation. ≫

```
2.29 {x0 → 0.843456, y0 → 0.0917789, z0 → 12.2494, T → 60.7365} -0.992139
2.291 {x0 → 0.843637, y0 → 0.0916503, z0 → 12.2494, T → 60.7713} -1.00104
2.292 {x0 → 0.8438, y0 → 0.0915315, z0 → 12.2494, T → 60.8061} -1.0099
2.293 {x0 → 0.843948, y0 → 0.091422, z0 → 12.2494, T → 60.841} -1.01873
2.294 {x0 → 0.844079, y0 → 0.0913216, z0 → 12.2494, T → 60.8759} -1.02752
2.295 {x0 → 0.844194, y0 → 0.0912302, z0 → 12.2494, T → 60.9109} -1.03628
2.296 {x0 → 0.844294, y0 → 0.0911474, z0 → 12.2494, T → 60.946} -1.04499
2.297 {x0 → 0.844379, y0 → 0.0910731, z0 → 12.2494, T → 60.9811} -1.05368
2.298 {x0 → 0.84445, y0 → 0.0910073, z0 → 12.2494, T → 61.0161} -1.06231
2.299 {x0 → 0.844506, y0 → 0.0909492, z0 → 12.2494, T → 61.0514} -1.07093
2.3 {x0 → 0.844548, y0 → 0.0908993, z0 → 12.2494, T → 61.0867} -1.0795
2.301 {x0 → 0.844577, y0 → 0.0908571, z0 → 12.2494, T → 61.1221} -1.08805
2.302 {x0 → 0.844591, y0 → 0.0908227, z0 → 12.2494, T → 61.1574} -1.09655
2.303 {x0 → 0.844593, y0 → 0.0907957, z0 → 12.2494, T → 61.1928} -1.105
2.304 {x0 → 0.844581, y0 → 0.0907761, z0 → 12.2494, T → 61.2283} -1.11343
2.305 {x0 → 0.844557, y0 → 0.0907638, z0 → 12.2494, T → 61.2638} -1.12181
2.306 {x0 → 0.844519, y0 → 0.0907587, z0 → 12.2494, T → 61.2994} -1.13016
2.307 {x0 → 0.844469, y0 → 0.0907606, z0 → 12.2494, T → 61.335} -1.13846
2.308 {x0 → 0.844407, y0 → 0.0907695, z0 → 12.2494, T → 61.3707} -1.14673
2.309 {x0 → 0.844332, y0 → 0.0907854, z0 → 12.2494, T → 61.4065} -1.15496
2.31 {x0 → 0.844245, y0 → 0.090808, z0 → 12.2494, T → 61.4423} -1.16315
```

```
ListPlot[Table[{b1, fm[b1]}, {b1, 2.27, 2.31, 0.001}]]
```