> $restart:$

Computation of a classical 2D bifurcation diagram for first-order coupling via singular ODEs

We use ideas from the Vessiot theory of singular differential equations to derive a vector field such that the bifurcation paths are trajectories and the bifurcation points steady states. In this worksheet, we always have N=M=3; all other parameters can be set in the worksheet. The bifurcation parameter is μ.

> $with(LinearAlgebra):with(plots):with(plottools):with(VectorCalculus):with(ListTools):with(RootFinding):with(Grid):with(ColorTools):$

> $interface(warnlevel=0);$
> $refinelvl := 1;$

$$3$$

$$refinelvl := 1 \tag{1}$$

> $tolbl := Color([68, 119, 170]):$
> $tolgr := Color([34, 136, 51]):$
> $tolye := Color([204, 187, 68]):$
> $tolpu := Color([170, 51, 119]):$
> $tolre := Color([238, 102, 119]):$
> $tolcy := Color([102, 204, 238]):$

Build model.

We set the parameters to their standard values in the paper and use a linear coupling coefficient function.

> $r := 0; o1 := 1; p1 := -1; q1 := 0; o2 := 1; p2 := -2; q2 := 3;$

$$r := 0$$
$$o1 := 1$$
$$p1 := -1$$
$$q1 := 0$$
$$o2 := 1$$
$$p2 := -2$$
$$q2 := 3 \tag{2}$$

> $s := (\mu, x) \rightarrow (o1 \cdot x^2 + p1 \cdot x + q1 - \mu) \cdot (o2 \cdot x^2 + p2 \cdot x + q2 - \mu);$
> $collect(s(\mu, x), x);$

$$s := (\mu, x) \mapsto (o1 \cdot x^2 + p1 \cdot x + q1 + (-\mu)) \cdot (o2 \cdot x^2 + p2 \cdot x + q2 + (-\mu))$$

$$x^4 - 3x^3 + (-2\mu + 5)x^2 + (3\mu - 3)x - \mu(-\mu + 3) \tag{3}$$

> $h := (\beta, \gamma, x) \rightarrow (\beta + \gamma \cdot x)$

$$h := (\beta, \gamma, x) \mapsto \beta + \gamma \cdot x \tag{4}$$

> $\sigma := y_1 + y_2 + y_3$

$$\sigma := y_1 + y_2 + y_3 \tag{5}$$

Define the considered vector field describing the evolution of the traits.

> $f_1 := (r \cdot \mu - y_1) \cdot expand(s(\mu, y_1)) + h(\beta, \gamma, y_1) \cdot \sigma;$

$f_2 := \left(r \cdot \mu - y_2\right) \cdot expand\left(s\left(\mu, y_2\right)\right) + h\left(\beta, \gamma, y_2\right) \cdot \sigma;$

$f_3 := \left(r \cdot \mu - y_3\right) \cdot expand\left(s\left(\mu, y_3\right)\right) + h\left(\beta, \gamma, y_3\right) \cdot \sigma;$

$$f_1 := -y_1 \left(y_1^4 - 2\,\mu\, y_1^2 - 3\, y_1^3 + \mu^2 + 3\,\mu\, y_1 + 5\, y_1^2 - 3\,\mu - 3\, y_1\right) + \left(\gamma y_1 + \beta\right)\left(y_1 + y_2 + y_3\right)$$

$$f_2 := -y_2 \left(y_2^4 - 2\,\mu\, y_2^2 - 3\, y_2^3 + \mu^2 + 3\,\mu\, y_2 + 5\, y_2^2 - 3\,\mu - 3\, y_2\right) + \left(\gamma y_2 + \beta\right)\left(y_1 + y_2 + y_3\right)$$

$$f_3 := -y_3 \left(y_3^4 - 2\,\mu\, y_3^2 - 3\, y_3^3 + \mu^2 + 3\,\mu\, y_3 + 5\, y_3^2 - 3\,\mu - 3\, y_3\right) + \left(\gamma y_3 + \beta\right)\left(y_1 + y_2 + y_3\right) \quad \textbf{(6)}$$

Compute its Jacobians both with respect to the variables only and in addition with respect to the environmental parameter μ.

> $J := simplify\left(Jacobian\left(\left[f_1, f_2, f_3\right], \left[y_1, y_2, y_3\right]\right)\right);$
> $JJ := simplify\left(Jacobian\left(\left[f_1, f_2, f_3\right], \left[\mu, y_1, y_2, y_3\right]\right)\right);$

$J :=$

$$\left[\begin{array}{cc} \cdots y_3)\,\gamma + \beta & \cdots \\ \cdots & -5\, y_2^4 + 12\, y_2^3 + 3\,(- \cdots \\ \cdots & \cdots \end{array}\right]$$

$JJ :=$ **(7)**

$$\left[\begin{array}{cc} 2\, y_1^3 - 3\, y_1^2 + \left(-2\,\mu + 3\right) y_1 & -5\, y_1^4 + 12\, y \cdots \\ 2\, y_2^3 - 3\, y_2^2 + \left(-2\,\mu + 3\right) y_2 & \cdots \\ 2\, y_3^3 - 3\, y_3^2 + \left(-2\,\mu + 3\right) y_3 & \cdots \end{array}\right]$$

> $detJ := Determinant(J):$

Setting up the vector field **Y** generating the **projected Vessiot distribution** using the adjoint of the Jacobian and the Jacobian with respect to μ alone.

> $C := Adjoint(J):$

> $M := Jacobian\left(\left[f_1, f_2, f_3\right], [\mu]\right);$

(8)

$$M := \begin{bmatrix} -y_1\left(-2\,y_1^2+2\,\mu+3\,y_1-3\right) \\ -y_2\left(-2\,y_2^2+2\,\mu+3\,y_2-3\right) \\ -y_3\left(-2\,y_3^2+2\,\mu+3\,y_3-3\right) \end{bmatrix} \quad \textbf{(8)}$$

> $b := -expand(C \cdot M)$:

> $Y := [detJ, b[1, 1], b[2, 1], b[3, 1]]$:

We consider now only the case β=1 and γ=0, i.e. pure first-order coupling.

> $bg01 := \beta = 1, \gamma = 0;$

$$bg01 := \beta = 1, \gamma = 0 \quad \textbf{(9)}$$

> $f01_1 := subs(bg01, f_1) : f01_2 := subs(bg01, f_2) : f01_3 := subs(bg01, f_3)$:
$J01 := subs(bg01, J) : JJ01 := subs(bg01, JJ) : detJ01 := subs(bg01, detJ)$:
$Y01 := subs(bg01, Y) : b01 := subs(bg01, b)$:

We use **Y** to set up a **system** in a form suitable for integration with **dsolve**.

> $vars := [\mu(t), y_1(t), y_2(t), y_3(t)]$:
$trafo := \mu = \mu(t), y_1 = y_1(t), y_2 = y_2(t), y_3 = y_3(t)$:

> $Y01t := subs(trafo, Y01)$:
$sysY01 := [diff(\mu(t), t) = Y01t[1], diff(y_1(t), t) = Y01t[2], diff(y_2(t), t) = Y01t[3],$
$diff(y_3(t), t) = Y01t[4]]$:

Compute numerically bifurcation points (requires larger number of digits for better identification of identical μ-values).

> $sys01 := [f01_1, f01_2, f01_3, detJ01]$:

> $st := time() : fsol01 := Isolate(sys01, [\mu, y_1, y_2, y_3], digits = 15) : time() - st; nops(fsol01);$

$$39.047$$

$$98 \quad \textbf{(10)}$$

We store the found bifurcation points in a **data file**.

> $fname$:= FileTools:-JoinPath(["Maple", "MathBiol", "Speciation" , cat("BifPointsN3M3B",
sprintf("%4.2f", subs(bg01, β)), ".txt")], base = homedir);
fd := fopen(fname, WRITE) :
fprintf (fd, "Bifurcation points computed by BifDia2DN3M3B1-Grid.mw\n") :
fprintf (fd,
"o1=%.1f, p1=%.1f, q1=%.1f, o2=%.1f, p2=%.1f, q2=%.1f, r=%.3f, N=%1d, beta=%4.2f\n",
o1, p1, q1, o2, p2, q2, r, 3, subs(bg01, β)) :
fprintf (fd, "mu\t\t x1\t\t x2\t\t x3\n") :

fname := "C:\Users\seiler\Maple\MathBiol\Speciation\BifPointsN3M3B1.00.txt" **(11)**

> **for** *bf* **in** *fsol01* **do**
fprintf (fd, "%+12.8f\t %+12.8f\t %+12.8f\t %+12.8f\n", op (map(rhs, bf))) :
end do:

```
> fclose(fd);
```

We select the μ-values of the bifurcation points and select those in the specified range.

```
> bifmu := MakeUnique(map(l→rhs(l[1]), fsol01)); nops(bifmu)
```

$bifmu := [-1.14719134232099, -0.791287847477920, -0.0310011227484499, -0.0310011227484500, 0., 1.27936991769510, 1.38931869536699, 1.96316081802837, 1.99549503414891, 2.24243649166109, 2.44844720885683, 2.57130488709475, 2.62365768427637, 2.78921053882435, 2.84458779275999, 2.91288544568714, 2.98506467728655, 3.00000000000000, 3.05831650338359, 3.07488076747505, 3.08446562047637, 3.37389058391727, 3.40192968845092, 3.69366927530896, 3.79128784747792, 3.87129972568736, 4.32043401368130, 4.34895585151540, 5.35670122007958, 5.38719927361829, 6.01344001561215, 6.01764549780501, 11.4344853132682, 14.4415582473110]$

$$34 \tag{12}$$

```
> μmin := -1; μmax := 7;
  bifmurel := select(x→(x > μmin) and (x < μmax), bifmu); nops(bifmurel);
  fsol01rel := select(x→rhs(x[1]) < μmax, fsol01) :
  bfpcolor := [ 0 $ nops(fsol01rel) ] :
```

$$\mu_{min} := -1$$

$$\mu_{max} := 7$$

$bifmurel := [-0.791287847477920, -0.0310011227484499, -0.0310011227484500, 0., 1.27936991769510, 1.38931869536699, 1.96316081802837, 1.99549503414891, 2.24243649166109, 2.44844720885683, 2.57130488709475, 2.62365768427637, 2.78921053882435, 2.84458779275999, 2.91288544568714, 2.98506467728655, 3.00000000000000, 3.05831650338359, 3.07488076747505, 3.08446562047637, 3.37389058391727, 3.40192968845092, 3.69366927530896, 3.79128784747792, 3.87129972568736, 4.32043401368130, 4.34895585151540, 5.35670122007958, 5.38719927361829, 6.01344001561215, 6.01764549780501]$

$$31 \tag{13}$$

Some auxiliary quantities for the plotting.

```
> proj12 := l→[l[1], Norm(Vector(l[2..4]))] :
  projsig := l→[l[1], l[2] + l[3] + l[4]] :
  round7 := x→1e-7·round(1e7·x) :
  box := [-1..7, -0.5..7];
  maxbox := [-2..300, -20..20];
  tend := 100;
```

$$box := [-1..7, -0.5..7]$$

$$maxbox := [-2..300, -20..20]$$

$$tend := 100 \tag{14}$$

We generate a list of μ-values lying before, after and between the chosen bifurcation values. The steady states at this μ-values are used as initial points for computing pieces of the bifurcation paths.

```
> nmu := nops(bifmurel);
  mulist := [ 0 $ nmu + 1 ]:
  checkmulist := [ 0 $ nmu + 1 ]:
```

$$nmu := 31 \tag{15}$$

```
> mulist[1] := bifmurel[1] - 1 :
  mulist[-1] := bifmurel[-1] + 1 :
> for i from 1 to nmu - 1 do
     mulist[i + 1] := 0.5·(bifmurel[i] + bifmurel[i + 1]);
  end do:
  checkmulist := [op(map(m→[μ(t) - m, halt], mulist)), [μ(t) - μ_max, halt]]:
```

The following procedure integrates Y along the i-th segment of the μ-list. For plotting the results in 2D, we use either the distance from the origin or the value of σ as ordinate.

```
> bifsegment := proc(i)
  local fsolm, ivps, intrange, checkmui, j, Jj, revsj, evsjm, evsjM, tsty, tcol, nvals, dsn, trajs, sigtrajs,
     sst;
  global nmu, tend, mulist, checkmulist, maxbox, refinelvl, J01, f01_1, f01_2, f01_3, trafo;
  fsolm := Isolate(subs(μ = mulist[i], [f01_1, f01_2, f01_3]), [y_1, y_2, y_3], Digits = 15);
  ivps := map(l→[μ(0) = mulist[i], op(l)], map(l→subs(trafo, t = 0, l), fsolm));
  if i = 1 then
     intrange := -tend..0;
  elif i = nmu + 1 then
     intrange := 0..tend;
  else
     intrange := -tend..tend;
  end if;
  trajs := Array(1..nops(fsolm), 0);
  sigtrajs := Array(1..nops(fsolm), 0);
  sst := Array(1..7, 0);
  checkmui := subsop(i = NULL, checkmulist);
  for j from 1 to nops(fsolm) do
     Jj := subs(op(ivps[j]), subs(trafo, t = 0, J01));
     revsj := map(Re, Eigenvalues(Jj));
     if mul(revsj) = 0 then
        tcol := "Black";
        tsty := solid;
        sst[6] := sst[6] + 1;
     end if;
     evsjM := max(revsj);
     evsjm := min(revsj);
     tsty := solid;
     nvals := nops({op(map(round7, map(rhs, fsolm[j])))});
     sst[nvals] := sst[nvals] + 1;
     if evsjM < 0 then
        tsty := solid;
```

```
        if nvals = 3 then
          tcol := tolgr;
          sst[7] := sst[7] + 1;
        else
          tcol := tolcy;
        end if;
        sst[4] := sst[4] + 1;
      elif evsjm > 0 then
        tcol := tolpu;
        tsty := dot;
        sst[5] := sst[5] + 1;
      elif evsjm·evsjM < 0 then
        tcol := tolye;
        tsty := dot;
        sst[5] := sst[5] + 1;
      else error "not yet handled";
      end if;
      dsn := dsolve([op(sysY01), op(ivps[j])], numeric, vars, method = rkf45, range = intrange,
       events = [[Y01t[1], halt], op(checkmui)]) :
      trajs[j] := odeplot(dsn, [μ(t), Norm(Vector([y_1(t), y_2(t), y_3(t)]))], color = tcol, linestyle
       = tsty, thickness = 1, refine = refinelvl, view = maxbox) :
      sigtrajs[j] := odeplot(dsn, [μ(t), y_1(t) + y_2(t) + y_3(t)], color = tcol, linestyle = tsty, thickness
       = 1, refine = refinelvl, view = maxbox) :
    end do:
    [i, nops(fsolm), trajs, sigtrajs, sst];
    end proc:
> Set('nmu','tend','mulist','checkmulist','maxbox','refinelvl','sysY01','Y01t','f01_1','f01_2','f01_3','J01','vars
     ','round7','trafo','tolbl','tolgr','tolye','tolpu','tolre','tolcy') :
> st := time[real]( ) : BifSegs := Map(bifsegment, [ii $ ii = 1 ..nmu + 1]) : time[real]( ) − st;
```

184.284 **(16)**

```
> trajs := Array(1 ..nmu + 1, 0) :
  sigtrajs := Array(1 ..nmu + 1, 0) :
  sst := Array(1 ..nmu + 1, 1 ..7, 0) :
  for i from 1 to nmu + 1 do
    print(BifSegs[i, 1], mulist[BifSegs[i, 1]], BifSegs[i, 2], BifSegs[i, 5]);
    trajs[i] := BifSegs[i, 3] :
    sigtrajs[i] := BifSegs[i, 4] :
    sst[i] := BifSegs[i, 5] :
  end do:
```

$$1, -1.791287848, 1, \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$2, -0.4111444851, 3, \begin{bmatrix} 3 & 0 & 0 & 2 & 1 & 0 & 0 \end{bmatrix}$$

$$3, -0.03100112275, 3, \begin{bmatrix} 3 & 0 & 0 & 2 & 1 & 0 & 0 \end{bmatrix}$$

$$4, -0.01550056138, 9, \begin{bmatrix} 3 & 6 & 0 & 2 & 7 & 0 & 0 \end{bmatrix}$$

$$5, 0.6396849590, 9, \begin{bmatrix} 3 & 6 & 0 & 2 & 7 & 0 & 0 \end{bmatrix}$$

$$6, 1.334344306, 15, \begin{bmatrix} 3 & 12 & 0 & 5 & 10 & 0 & 0 \end{bmatrix}$$

$$7, 1.676239756, 21, \begin{bmatrix} 3 & 12 & 6 & 5 & 16 & 0 & 0 \end{bmatrix}$$

$$8, 1.979327926, 33, \begin{bmatrix} 3 & 12 & 18 & 5 & 28 & 0 & 0 \end{bmatrix}$$

$$9, 2.118965763, 27, \begin{bmatrix} 3 & 12 & 12 & 5 & 22 & 0 & 0 \end{bmatrix}$$

$$10, 2.345441850, 33, \begin{bmatrix} 3 & 18 & 12 & 5 & 28 & 0 & 0 \end{bmatrix}$$

$$11, 2.509876048, 39, \begin{bmatrix} 3 & 18 & 18 & 5 & 34 & 0 & 0 \end{bmatrix}$$

$$12, 2.597481286, 45, \begin{bmatrix} 3 & 24 & 18 & 5 & 40 & 0 & 0 \end{bmatrix}$$

$$13, 2.706434112, 51, \begin{bmatrix} 3 & 24 & 24 & 5 & 46 & 0 & 0 \end{bmatrix}$$

$$14, 2.816899166, 45, \begin{bmatrix} 3 & 18 & 24 & 2 & 43 & 0 & 0 \end{bmatrix}$$

$$15, 2.878736620, 57, \begin{bmatrix} 3 & 18 & 36 & 8 & 49 & 0 & 6 \end{bmatrix}$$

$$16, 2.948975062, 69, \begin{bmatrix} 3 & 18 & 48 & 8 & 61 & 0 & 6 \end{bmatrix}$$

$$17, 2.992532338, 71, \begin{bmatrix} 5 & 18 & 48 & 9 & 62 & 0 & 6 \end{bmatrix}$$

$$18, 3.029158252, 71, \begin{bmatrix} 5 & 18 & 48 & 9 & 62 & 0 & 6 \end{bmatrix}$$

$$19, 3.066598635, 65, \begin{bmatrix} 5 & 12 & 48 & 9 & 56 & 0 & 6 \end{bmatrix}$$

$$20, 3.079673194, 71, \begin{bmatrix} 5 & 18 & 48 & 9 & 62 & 0 & 6 \end{bmatrix}$$

$$21, 3.229178102, 71, \begin{bmatrix} 5 & 18 & 48 & 9 & 62 & 0 & 6 \end{bmatrix}$$

$$22, 3.387910136, 77, \begin{bmatrix} 5 & 24 & 48 & 12 & 65 & 0 & 6 \end{bmatrix}$$

$$23, 3.547799482, 83, \begin{bmatrix} 5 & 24 & 54 & 12 & 71 & 0 & 6 \end{bmatrix}$$

$24, 3.742478561, 89, \begin{bmatrix} 5 & 30 & 54 & 15 & 74 & 0 & 6 \end{bmatrix}$

$25, 3.831293786, 89, \begin{bmatrix} 5 & 30 & 54 & 15 & 74 & 0 & 6 \end{bmatrix}$

$26, 4.095866870, 95, \begin{bmatrix} 5 & 36 & 54 & 18 & 77 & 0 & 6 \end{bmatrix}$

$27, 4.334694933, 101, \begin{bmatrix} 5 & 42 & 54 & 18 & 83 & 0 & 6 \end{bmatrix}$

$28, 4.852828536, 101, \begin{bmatrix} 5 & 42 & 54 & 18 & 83 & 0 & 6 \end{bmatrix}$

$29, 5.371950245, 107, \begin{bmatrix} 5 & 48 & 54 & 21 & 86 & 0 & 6 \end{bmatrix}$

$30, 5.700319645, 113, \begin{bmatrix} 5 & 48 & 60 & 21 & 92 & 0 & 6 \end{bmatrix}$

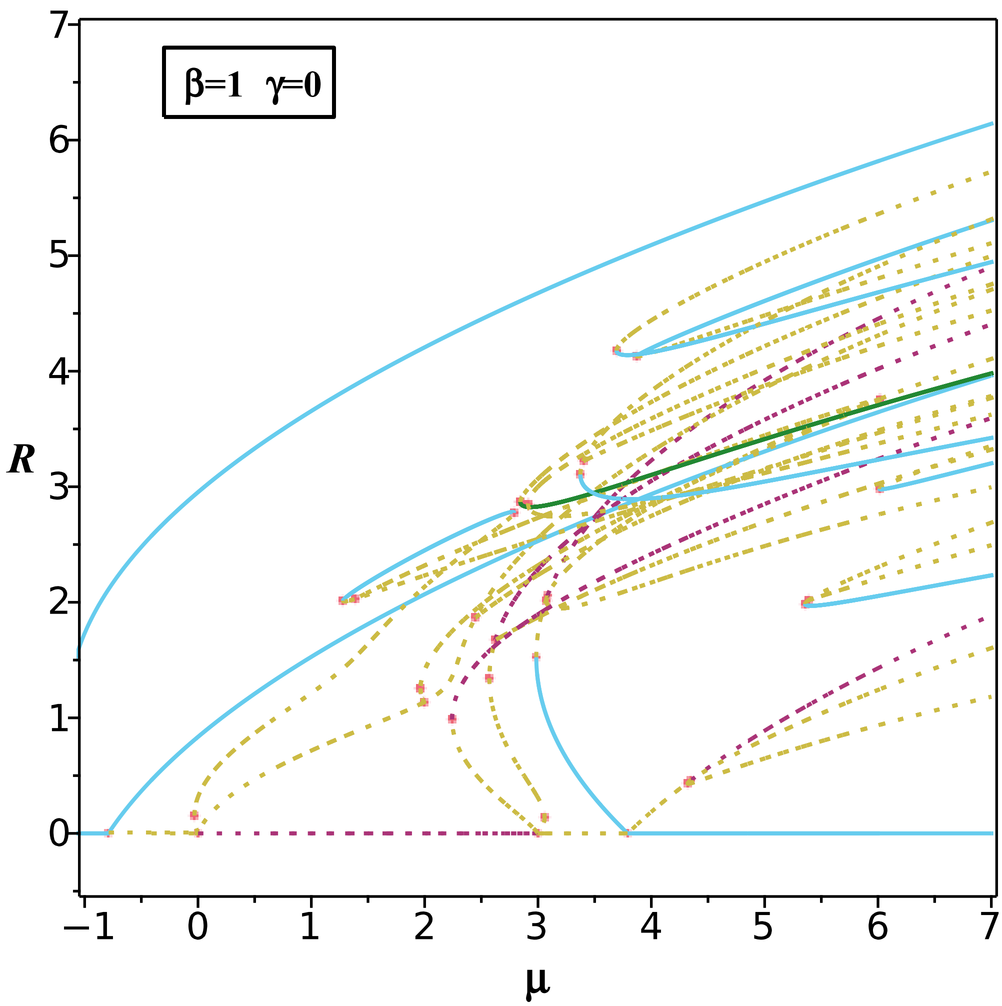
$31, 6.015542755, 119, \begin{bmatrix} 5 & 54 & 60 & 24 & 95 & 0 & 6 \end{bmatrix}$

$32, 7.017645498, 107, \begin{bmatrix} 5 & 54 & 48 & 24 & 83 & 0 & 6 \end{bmatrix}$ **(17)**

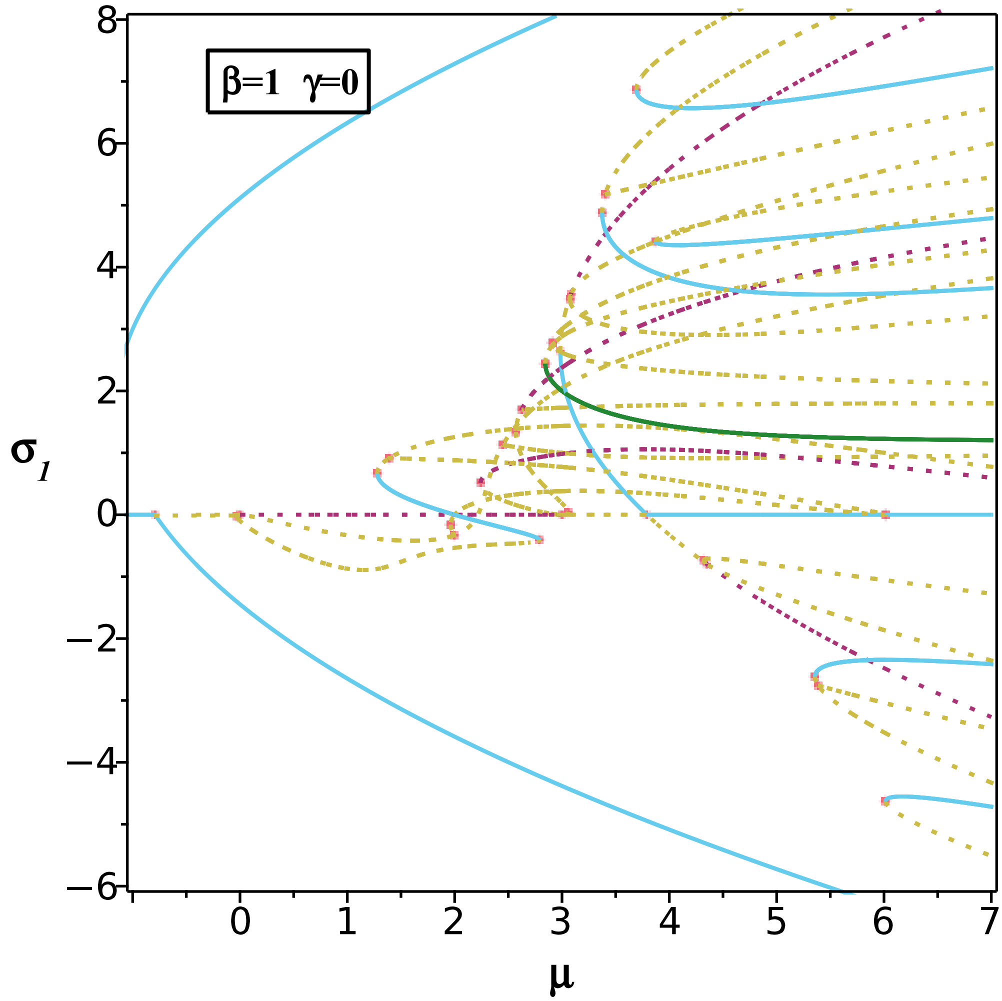
```
> add(ArrayNumElems(trajs[i]), i = 1 ..ArrayNumElems(trajs))
```

1890 **(18)**

```
> bifs := map(l→proj12(map(rhs, l)), fsol01rel) :
  bifsig := map(l→projsig(map(rhs, l)), fsol01rel) :
  bifsp := pointplot(bifs, symbol = soliddiamond, color = tolre, symbolsize = 7) :
  bifsigp := pointplot(bifsig, symbol = soliddiamond, color = tolre, symbolsize = 7) :
> display(bifsp, seq(seq(trajs[i][j], j = 1 ..ArrayNumElems(trajs[i])), i = 1
     ..ArrayNumElems(trajs)),
         textplot([0.5, 6.5, typeset("β=", subs(bg01, β), " γ=", subs(bg01, γ)), font
     = [Helevetica, bold, 14]]),
         rectangle([−0.3, 6.2], [1.2, 6.8], filled = false), font = [Helvetica, roman, 14],
         legendstyle = [location = top, font = [Helvetica, bold, 14]],
         labels = ['μ','R'], labelfont = [Helevetica, bold, 16], view = box, axes = boxed)
```



> $display(bifsigp, seq(seq(sigtrajs[i][j], j = 1 ..ArrayNumElems(trajs[i])), i = 1 ..ArrayNumElems(trajs)),$
> $textplot([0.5, 7, typeset(\text{"β=", } subs(bg01, \beta), \text{" γ=", } subs(bg01, \gamma)), font = [Helevetica, bold, 14]]),$
> $rectangle([-0.3, 6.5], [1.2, 7.5], filled = false), font = [Helvetica, roman, 14],$
> $labels = [\text{'μ','}\sigma_1\text{'}], labelfont = [Helevetica, bold, 16], view = [-1 ..7, -6 ..8], axes = boxed)$



β=1 γ=0
σ_1
μ
8
6
4
2
0
−2
−4
−6
0
1
2
3
4
5
6
7

>