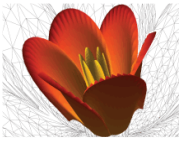


# IMS 2008 June 20-24th Internation

Spectra and Bifurcation Diagrams on the Web

drawing the zero contour of a function with a minimal number of evaluations

Jiří Benedikt

Katedra matematiky
Západočeská univerzita v Plzni
Univerzitní 22
306 14 Plzeň
Czech Republic

Introduction

Many types of spectra and bifurcation diagrams of (not only) boundary value problems for ordinary d Evaluation of such function may be very slow, especially in the case of nonlinear higher-order differ *spectrum* of the *fourth-order Dirichlet boundary value problem* (0)

$$u^{(4)}(t) = \mu u^{+}(t) - \nu u^{-}(t), \qquad t \in [0, 1],$$
$$u(0) = u'(0) = u(1) = u'(1) = 0,$$

where $u^{+} = \max\{u, 0\}$ and $u^{-} = \max\{-u, 0\}$, i.e. $u = u^{+} - u^{-}$. The Fučík spectrum is defined as the s the oscillation of a suspension bridge is related to the shape of the Fučík spectrum of (0).

The author has been supported by the Grant 1N04078 and by the Research Plan MSM 4977751301 c 177436.

Mathematical Model of a Suspension Bridge and the Fučík Spectrum

There are many mathematical models of suspension bridge, at different levels of simplification (see

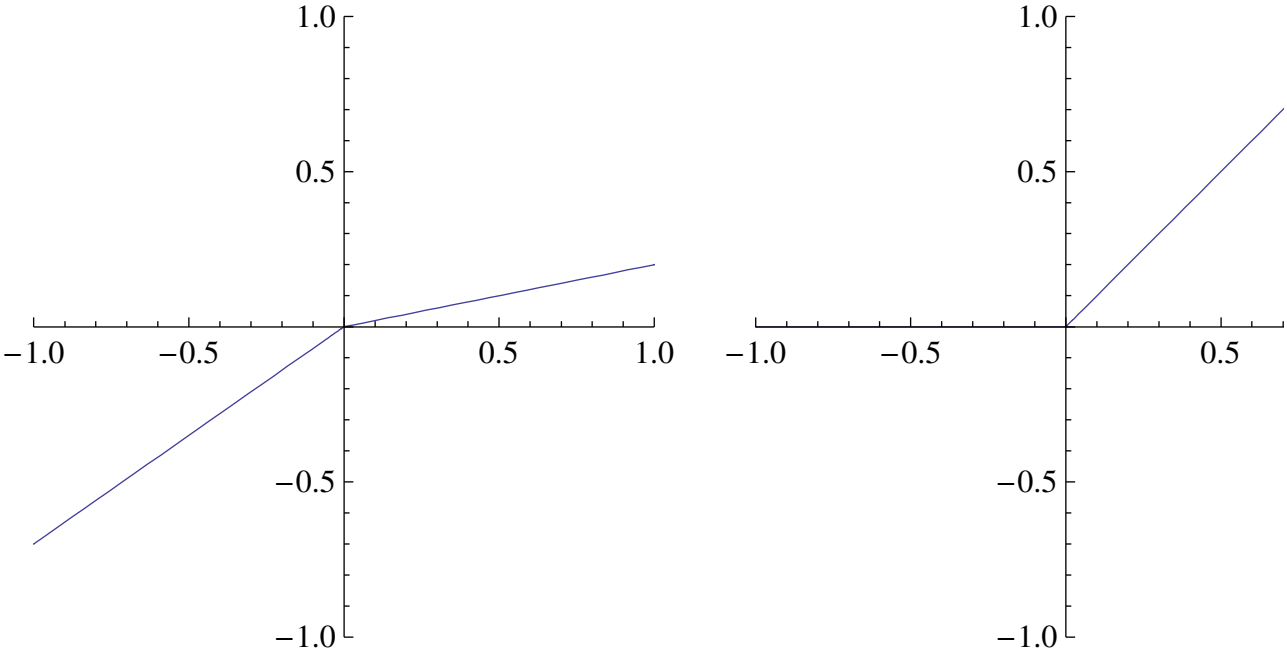
$$u^{(4)}(t) = f(u(t)), \qquad t \in [0, 1],$$
$$u(0) = u'(0) = u(1) = u'(1) = 0,$$

where $u : [0, 1] \to \mathbb{R}$ (the solution) stands for the deflection of the bridge deck and $f : \mathbb{R} \to \mathbb{R}$ describ bears on the stationary solution of a less simplified initial–boundary value problem (cf. [0]).

The external force $f$ depends on the deflection $u$, as it is typical for the suspension bridge — if the br ($u < 0$), the cable stays do nothing.

Let us assume that $f$ depends linearly on $u$, but with different constants for $u$ positive and $u$ negative, stays behave like a spring with the modulus $\mu$ for $u$ positive and the modulus $\nu$ for $u$ negative ($\nu = 0$

```
PosPart[a_] := (1 / 2) * (a + Abs[a]);
NegPart[a_] := (1 / 2) * (Abs[a] - a);
GraphicsRow[Plot[#1[[1]] * PosPart[t] - #1[[2]] * NegPart[t], {t, -1, 1}, AspectRatio → Automatic,
```




We arrive at the fourth-order Dirichlet boundary value generalized eigenvalue problem (0)

$u^{(4)}(t) = \mu u^{+}(t) - \nu u^{-}(t), \quad t \in [0, 1],$
$u(0) = u'(0) = u(1) = u'(1) = 0,$

The Fučík (generalized) spectrum of (0) is defined as the set of all couples $(\mu,\nu) \in \mathbb{R}^2$ such that (0) h

Shooting Method

Let us consider the fourth-order initial value problem (0)

$u^{(4)}(t) = \mu u^{+}(t) - \nu u^{-}(t), \quad t \in [0, 1],$
$u(0) = u'(0) = 0,\ u''(0) = 1,\ u'''(0) = \delta,$

where $\mu, \nu > 0$ and $\delta \in \mathbb{R}$. The solution can be proved to be unique, and so we may denote by V = V exactly one constant $\delta_0 = \delta_0(\mu, \nu)$, such that $V(\mu, \nu, \delta_0(\mu, \nu)) = 0$. Since clearly $V(\mu, \nu, \delta) > 0$ for an $(\mu, \nu)$ belongs to the Fučík spectrum of (0). Moreover, $(\nu, \mu)$ belongs to the Fučík spectrum of (0), tc

On the other hand, let us assume that $(\mu, \nu)$, $\mu, \nu > 0$, belongs to the Fučík spectrum of (0) and let u l suppose that u''(0) = 1. Then, clearly, u is the solution of (0) with $\delta = \delta_0(\mu, \nu)$, and so $D(\mu, \nu) = 0$. If

We showed that $(\mu, \nu)$, $\mu, \nu > 0$, belongs to the Fučík spectrum of (0), **if and only if** $D(\mu, \nu) = 0$ or I

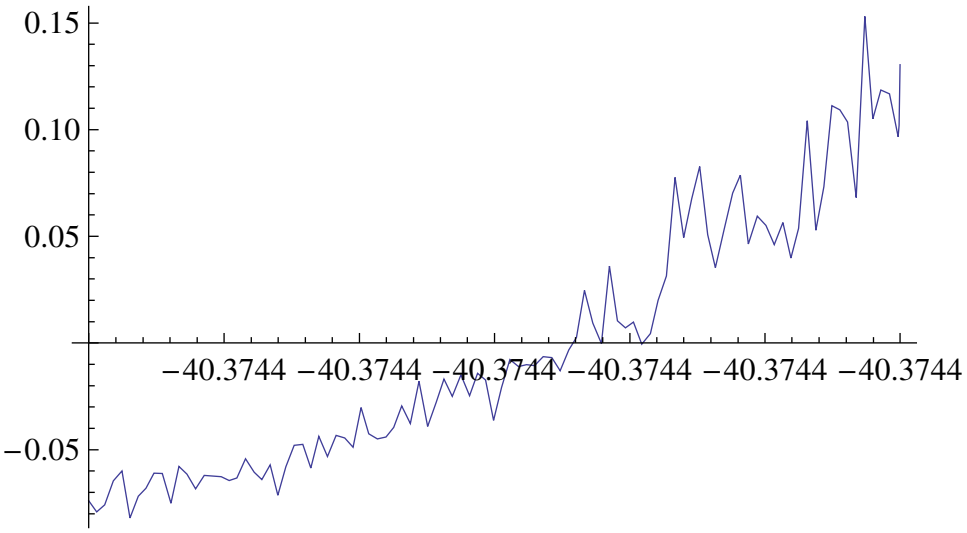
Implementation

To get a reasonable picture of the Fučík spectrum, we first replace $\mu$ and $\nu$ by $\mu^4$ and $\nu^4$ in (0), respec define the function V using the `NDSolve` method with the implicit machine working precision:

```
ValueAtOne[μ_?NumericQ, ν_?NumericQ, δ_?NumericQ] :=
    Block[{t}, u[1] /. NDSolve[{Derivative[4][u][t] == μ^4 * PosPart[u[t]] - ν^4 * NegPart[u[t]], u[0
```

we cannot find $\delta_0$ with sufficient precision:

Plot[ValueAtOne[40, 30, $\delta$], {$\delta$, −40.37441848174, −40.37441848177}]




We have to increase the working precision, but we realize that the solution is then not computed on t

```
ValueAtOne[μ_?NumericQ, ν_?NumericQ, δ_?NumericQ, goal_?NumericQ, working_?Numeri
    Block[{t}, u[1] /. NDSolve[{Derivative[4][u][t] == SetPrecision[μ^4, working] * PosPart[u[t]] − S
          Derivative[2][u][0] == 1, Derivative[3][u][0] == SetPrecision[δ, working]}, u, {t, 0, 1}, I
TableForm[Prepend[Table[{2 * n, ValueAtOne[40, 30, −40.37441848174, n, 2 * n]}, {n, 20, 30}], {w
```

NDSolve::nderr : Error test failure at t == 0.279833937231701600345094946814981749601235472

InterpolatingFunction::dmval : Input value {1} lies outside the range of data in the interpolating func

NDSolve::nderr : Error test failure at t == 0.209036355469240029154026729522888077061225124

InterpolatingFunction::dmval : Input value {1} lies outside the range of data in the interpolating func

NDSolve::nderr : Error test failure at t == 0.569973935081085825042185038237627927430426952

General::stop : Further output of NDSolve::nderr will be suppressed during this calculation. ≫

InterpolatingFunction::dmval : Input value {1} lies outside the range of data in the interpolating func

General::stop : Further output of InterpolatingFunction::dmval will be suppressed during this calcula

| working | result |
|---|---|
| 40 | 4149.200783168546420504861145407557206 9 |
| 42 | 4149.200783183007774869332316939742606669 |
| 44 | 4149.200783185747382804345199575296665462 39 |
| 46 | 4149.200783186302074197083841648580444186624 0 |
| 48 | 4149.200783186270899750961555627010355626238360 |
| 50 | $-2.9400489100191658701 \times 10^{21}$ |
| 52 | $-5.837053188698064 \times 10^{18}$ |
| 54 | $-3.32962670635164491687 \times 10^{24}$ |

| | |
|---|---|
| 56 | $9.37495264008571522908005 5 \times 10^{6}$ |
| 58 | 243.21733086442708650614000 72 |
| 60 | $9.673256282136531403288 5 \times 10^{33}$ |

As in *Mathematica* 5.2.2.0, this problem can be fixed using the **StiffnessSwitching** method.

```
ValueAtOne[order_?NumericQ, μ_?NumericQ, ν_?NumericQ, δ_?NumericQ, goal_?NumericQ
    Block[{t}, Derivative[order][u][1] /. NDSolve[{Derivative[4][u][t] == SetPrecision[μ^4, working
            Derivative[2][u][0] == 1, Derivative[3][u][0] == SetPrecision[δ, working]}, u, {t, 0, 1}, I
TableForm[Prepend[Table[{2 * n, ValueAtOne[0, 40, 30, -40.37441848174, n, 2 * n]}, {n, 20, 30}],
```

| working | result |
|---|---|
| 40 | 4149.200783187694578011346440309851041 0 |
| 42 | 4149.200783186514877811220844928897271843 |
| 44 | 4149.200783186318852314056873364127266 07 |
| 46 | 4149.200783186327086275132659193660971361 7054 |
| 48 | 4149.200783186326148468775076447726268469326921 |
| 50 | 4149.200783186326162533582024750592002669616302 96 |
| 52 | 4149.200783186326127164305608177276405932198294 8513 |
| 54 | 4149.200783186326129272169001134865272672561646468522 |
| 56 | 4149.200783186326129121884854843244462994417499143215 58 |
| 58 | 4149.200783186326129115688037355424030426696700844048414 6 |
| 60 | 4149.200783186326129116716870787972247786475578421119248786 |

In *Mathematica* 5.2.2.0, we obtained results that still did not converge to a value due to non-smoothn not need to stop the integration at points where the solution changes its sign as we did in *Mathematic*

```
DerZero[μ_?NumericQ, ν_?NumericQ] := ValueAtOne[1, μ, ν, δ /. FindRoot[ValueAtOne[0, μ, ν
```

Avoiding the event location allows us to use lower working precision that in *Mathematica* 5.2.2.0. M point may still attain a minute (running on Intel Core 2 Duo 1.8 GHz with 1GB RAM):

```
DerZero[40, 30] // Timing
```

```
{44.351, -0.021774825060069010027294306758102588603139198806 3}
```

Optimization

To draw the zero countour of D in a reasonable time, we use an algorithm introduced in [1]. Here we

```
ZLInit[f_, xmin_?NumericQ, xmax_?NumericQ, ymin_?NumericQ, ymax_?NumericQ, proporre
    Module[{state = {{f, xmin, xmax, ymin, ymax, proporref, 0, 0, 0, 0}, Table[0, {2}, {2}], Table[0,
ZLAddLev[state_] := Module[{ls = state}, ls[[1, 8]]++;
    ls[[2]] = ls[[2]] /. vec_?VectorQ :> Drop[Flatten[Transpose[{vec, Table[0, {2^(ls[[1, 8]] - 1)
    ls[[2]] = Drop[Flatten[Transpose[{ls[[2]], Table[0, {2^(ls[[1, 8]] - 1) + 1}, {2^ls[[1, 8]] + 1}]}]
    ls[[3]] = ls[[3]] /. vec_?VectorQ :> Drop[Flatten[Transpose[{vec, Table[0, {2^(ls[[1, 8]] - 1)
    ls[[3]] = Drop[Flatten[Transpose[{ls[[3]], Table[0, {2^(ls[[1, 8]] - 1) + 1}, {2^ls[[1, 8]] + 1}]}]
    ls];
ZLRefDiv[state_, lev_?IntegerQ] := Module[{dif, indeces},
    dif = 2^(state[[1, 8]] - lev);
```

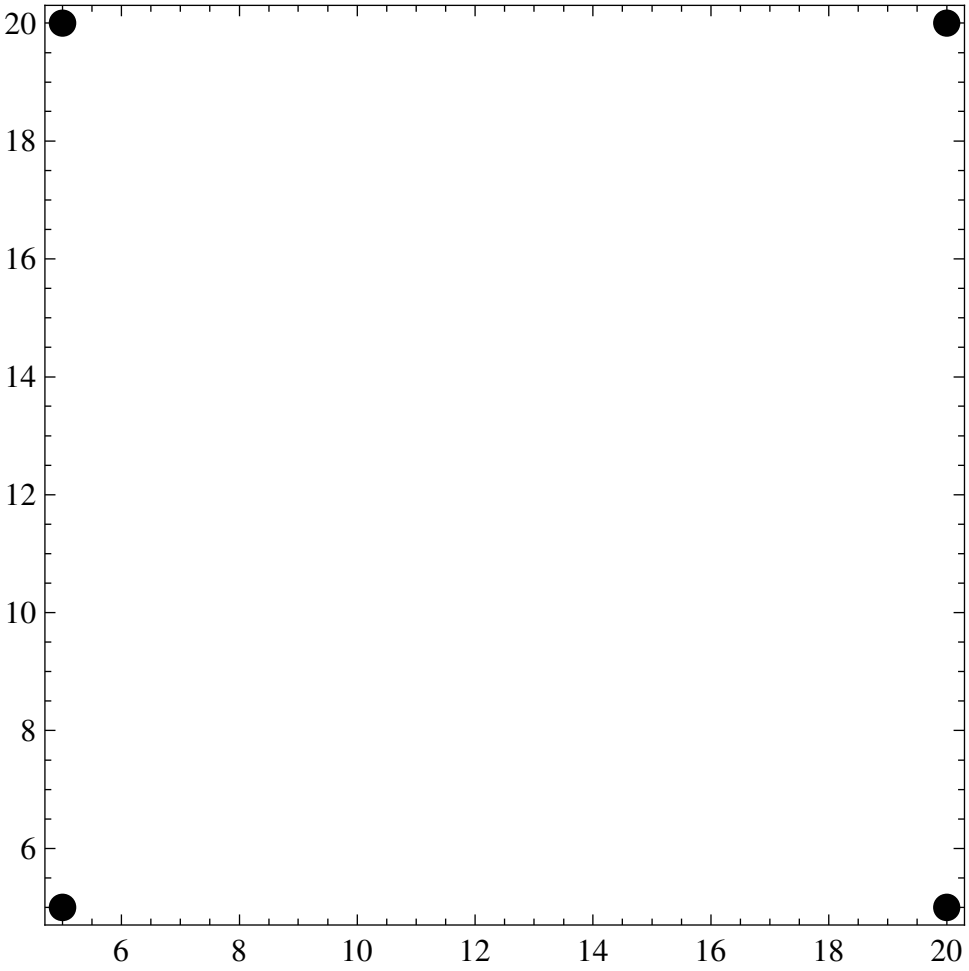
```
    indeces = Flatten[Table[Select[Flatten[Table[{i * dif + 1, j * dif + 1}, {j, dir, 2^lev - dir, 2}, {i,
        state[[3, #[[2]] + If[dir == 0, 0, {-1, 0, 1} * dif], #[[1]] + If[dir == 0, {-1, 0, 1} * dif, 0]
    ZLEval[state, indeces, False]];
ZLRefNei[state_, lev_?IntegerQ] := Module[{dif, pos, indeces},
    dif = 2^(state[[1, 8]] - lev);
    pos = Flatten[{{#, Map[Reverse, #, {2}]} &[Flatten[Table[{{1 * (-1)^j, 0}, {1 * (-1)^j, 1 * (-1
    indeces = Select[Flatten[Table[{i * dif + 1, j * dif + 1}, {j, 0, 2^lev}, {i, 0, 2^lev}], 1], Functio
        (Table[state[[{2, 3}, #[[i, 2]], #[[i, 1]]]], {i, 1, 2}] & /@ Select[Table[{#, #} &[{i
    ZLEval[state, indeces, False]];
ZLUni[state_] := Module[{ls = state, lev, dif, indeces},
    lev = ++ls[[1, 7]];
    dif = 2^(ls[[1, 8]] - lev);
    indeces = Select[Flatten[Table[{i * dif + 1, j * dif + 1}, {dir, 0, 1}, {j, dir, 2^lev - dir, 2}, {i, 1 -
    ZLEval[ls, indeces, True]];
ZLEval[state_, ind_, uni_] :=
  Module[{ls = state}, ({ls[[2, #[[2]], #[[1]]]], ls[[3, #[[2]], #[[1]]]]} = {ls[[1, 1]][ls[[1, 2]] + (ls[[
    If[uni, ls[[1, 9]] = (2^ls[[1, 7]] + 1)^2; If[ls[[1, 7]] > 0, ls[[1, 10]] -= ((2^ls[[1, 7]] + 1)^2 -
    ls];
ZLRefine[state_] := Module[{ls = state}, For[i = state[[1, 7]] + 1, i ≤ state[[1, 8]], i++, ls = ZLRefI
ZLRefineRep[state_] := Module[{ls = state, pts}, pts = Plus @@ ls[[1, {9, 10}]]; ls = ZLRefine[ls]; W
ZLUnEval[values_, bools_] := Module[{ls = values, maxi, val, fi},
    maxi = Length[ls[[1]]];
    For[i = 1, i ≤ maxi, i++, val = Null;
      For[j = 1, j ≤ maxi, j++, If[bools[[i, j]] == 1, If[val === Null, fi = j]; val = ls[[i, j]], ls[[i, j
      If[val =!= Null, val = ls[[i, fi]]; For[j = fi - 1, j ≥ 1, j--, ls[[i, j]] = val;];];];
    For[j = 1, j ≤ maxi, j++, val = Null; For[i = 1, i ≤ maxi, i++, If[ls[[i, j]] =!= Null, If[val ==
      val = ls[[fi, j]]; For[i = fi - 1, i ≥ 1, i--, ls[[i, j]] = val;];];
    ls];
ZLUnEvalCom = Compile[{{values, _Real, 2}, {bools, _Integer, 2}}, ZLUnEval[values, bools]];
ZLStep[state_] := Module[{ls = state, ifrefine, pts}, pts = Plus @@ ls[[1, {9, 10}]]; ifrefine = ls[[1, 1(
    If[! ifrefine || ls[[1, 10]] == 0, ls = ZLUni[ls]; ls = ZLRefineRep[ls]; ls[[2]] = ZLUnEvalCom[l
    ls];
ZLShow[state_, rules_] :=
  ListContourPlot[Flatten[Table[{state[[1, 2]] + i * 2^(-state[[1, 8]]) (state[[1, 3]] - state[[1, 2]]), sta
    Contours → {0}, ContourShading → False, rules, PlotRange → {Take[state[[1]], {2, 3}], Take[stat
ZLShow[state_] := ZLShow[state, {}]
ZLShowSym[state_] := Show[ZLShow[state], ZLShow[{state[[1]], Transpose[state[[2]]], state[[3]]}
ZLShowPts[state_, ptsize_, rules_] :=
  Show[ZLShow[state, DisplayFunction → Identity], Graphics[{PointSize[ptsize], Point[{state[[1,
      Select[Flatten[Table[{i, j}, {i, 0, 2^state[[1, 8]]}, {j, 0, 2^state[[1, 8]]}], 1], state[[3, #[[
    rules, PlotRange → ({{1.02, -0.02}, {-0.02, 1.02}}.# & /@ {Take[state[[1]], {2, 3}], Take[state[
ZLShowPts[state_, ptsize_] := ZLShowPts[state, ptsize, {}];
ZLInfo[state_] := Block[{evals = Plus @@ state[[1, {9, 10}]], allc = (2^state[[1, 8]] + 1)^2}, ToStrin
```

Since the time of evaluation of D increases with $\mu$ and $\nu$, we demontrate the algorithm on drawing th
`ZLStep` on the previous state until we are contented with the result. The result is shown using the `Z`
means that the algorithm refines the curves of the zero contour that it has already found only if less t

```
total = Timing[state = ZLInit[DerZero, 5., 20., 5., 20., 0.9]][[1]]
ZLInfo[state]
ZLShowPts[state, 0.03]
```

12.043

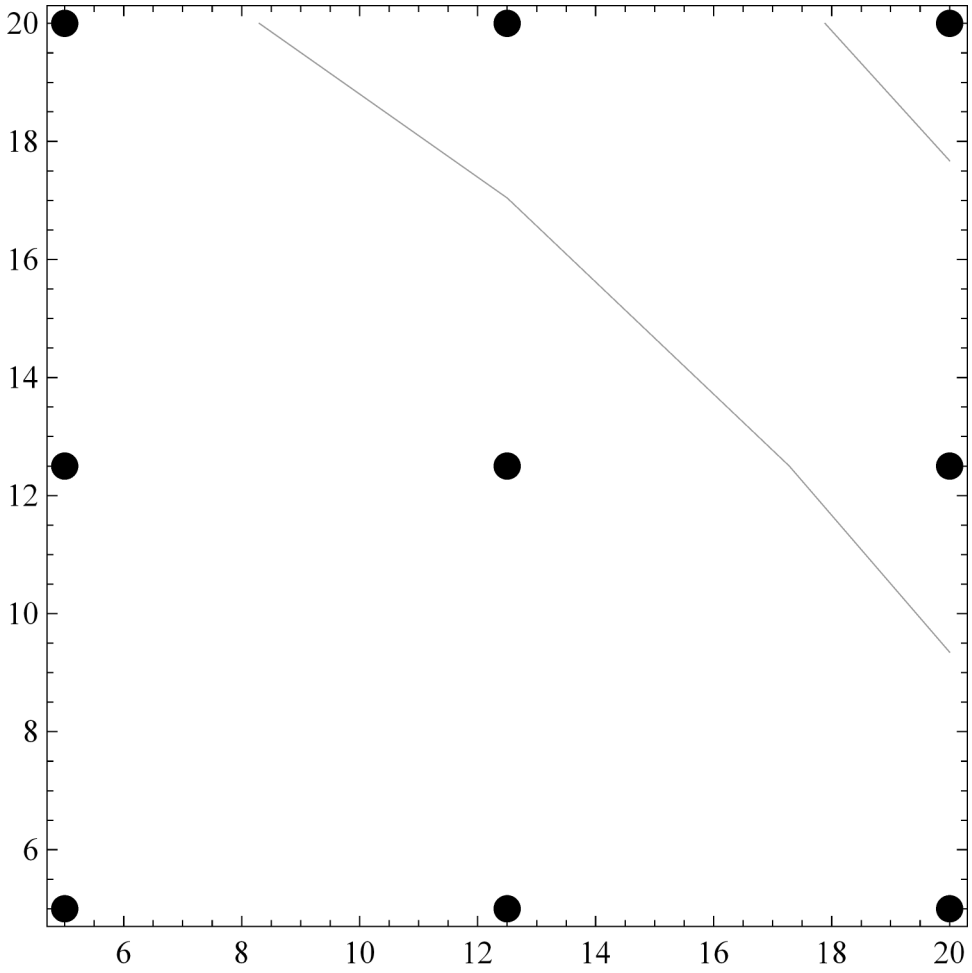
{0, 0, 4, 0}, 4 / 4 (100.%)




```
total += Timing[state = ZLStep[state]][[1]]
ZLInfo[state]
ZLShowPts[state, 0.03]
```

43.04

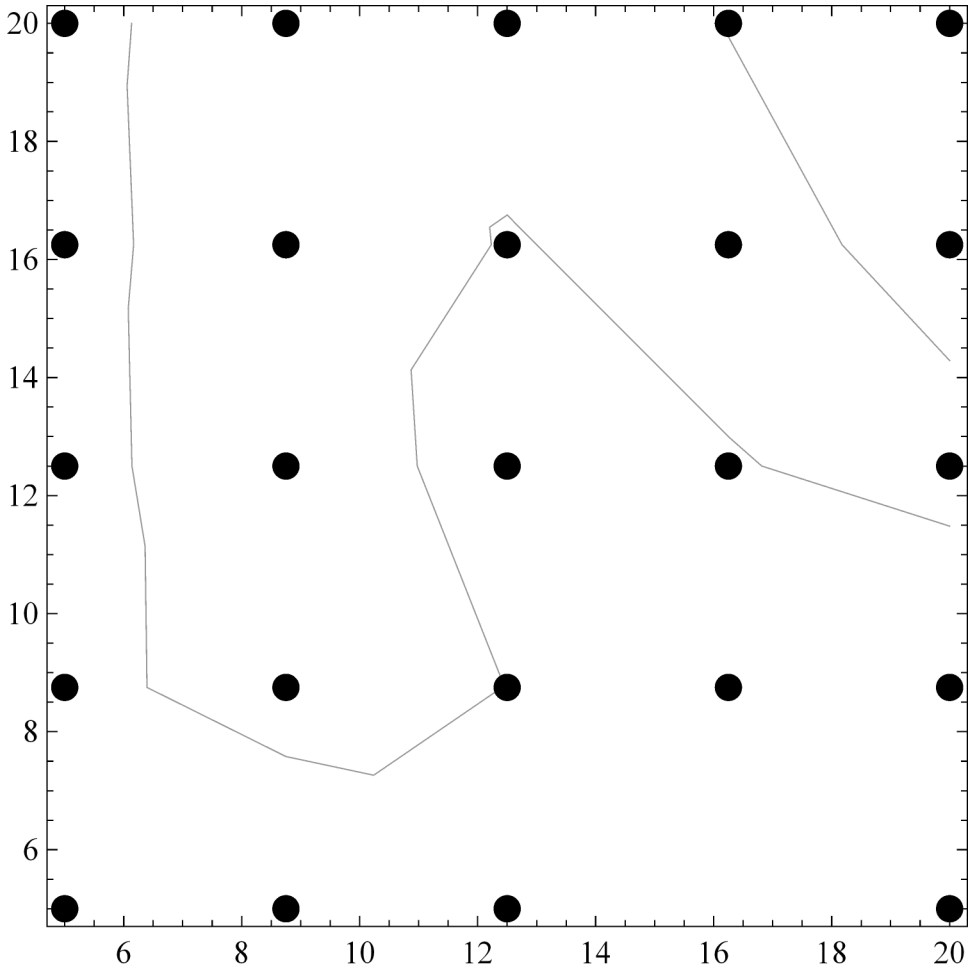
{1, 1, 9, 0}, 9 / 9 (100.%)




```
total += Timing[state = ZLStep[state]][[1]]
ZLInfo[state]
ZLShowPts[state, 0.03]
```

149.792

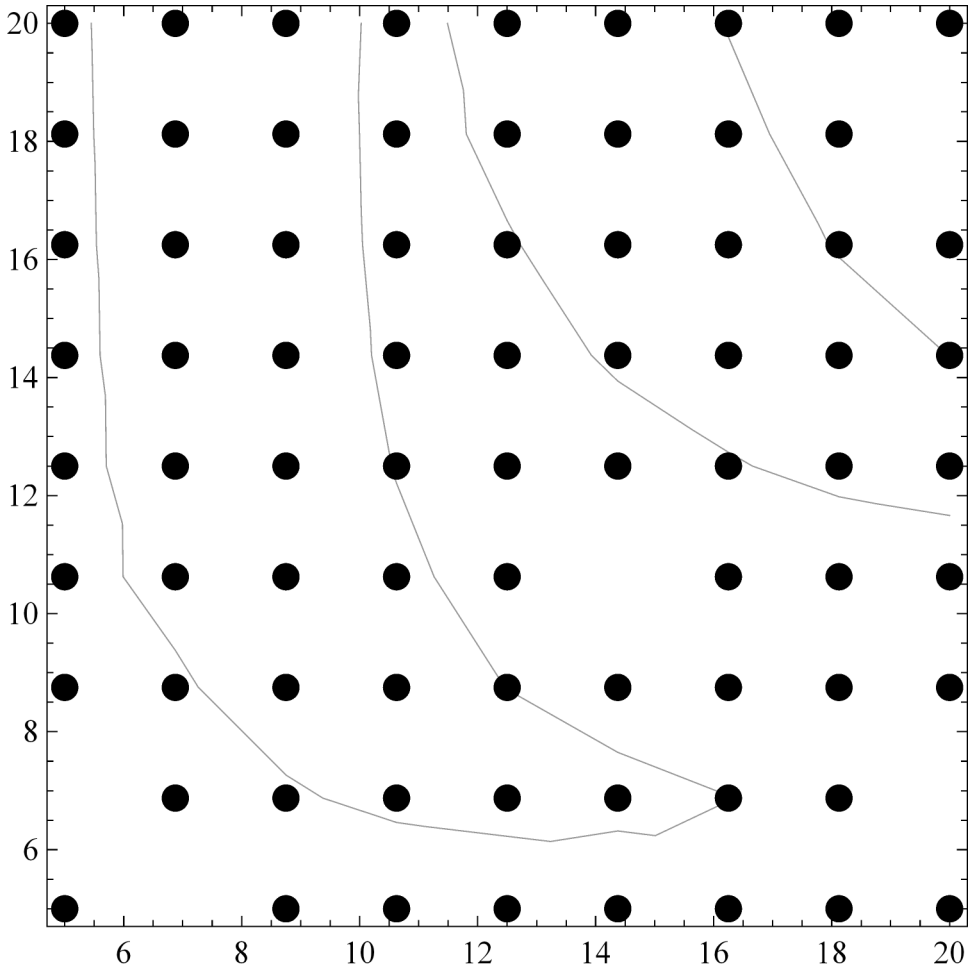
{1, 2, 9, 15}, 24 / 25 (96.%)




```
total += Timing[state = ZLStep[state]][[1]]
ZLInfo[state]
ZLShowPts[state, 0.03]
```

539.326

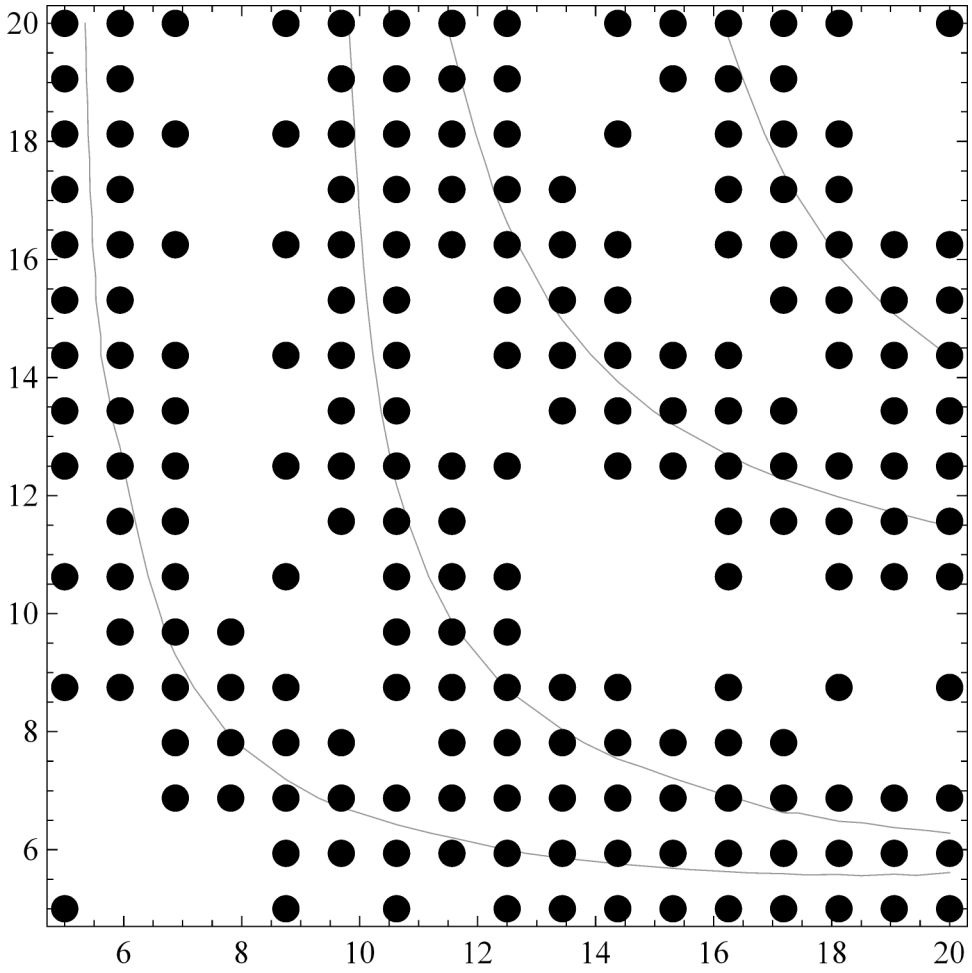
{1, 3, 9, 67}, 76 / 81 (93.8272%)




```
total += Timing[state = ZLStep[state]][[1]]
ZLInfo[state]
ZLShowPts[state, 0.03]
```

1481.59

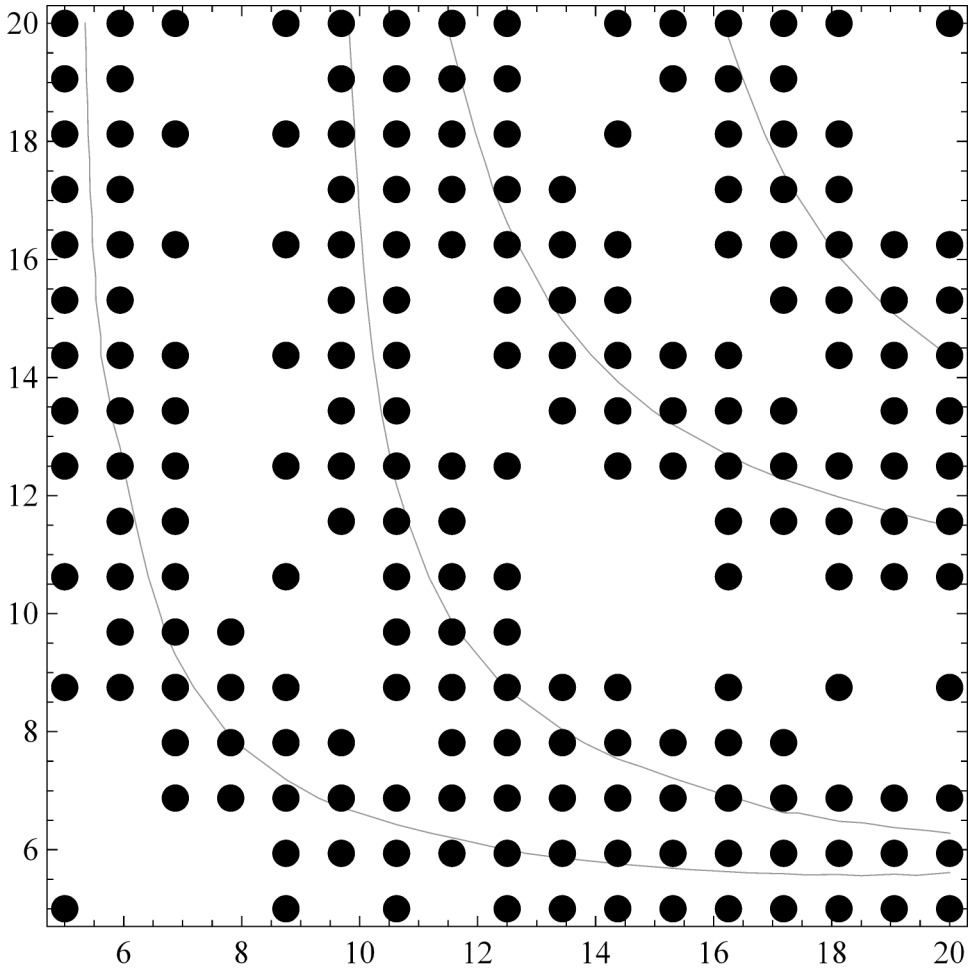
{1, 4, 9, 194}, 203 / 289 (70.2422%)




```
total += Timing[state = ZLStep[state]][[1]]
ZLInfo[state]
ZLShowPts[state, 0.03]
```

1481.63

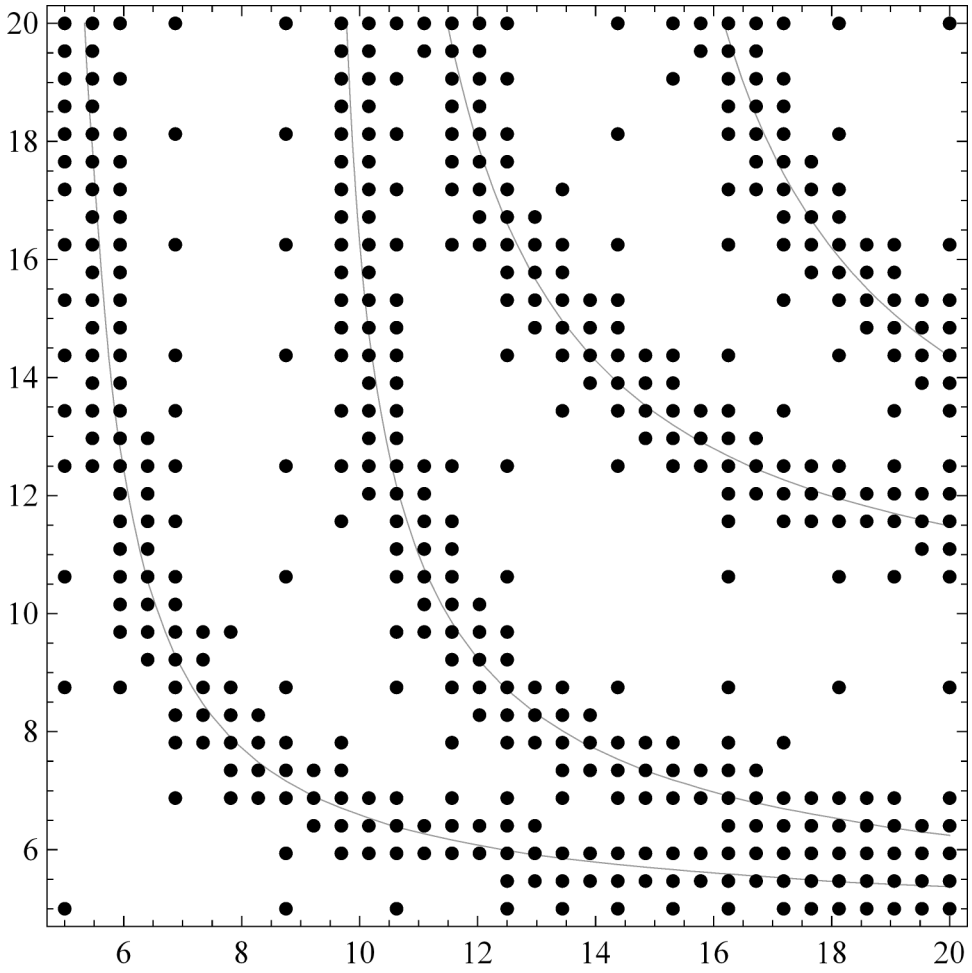
{2, 4, 25, 178}, 203 / 289 (70.2422%)




total += Timing[state = ZLStep[state]][[1]]
ZLInfo[state]
ZLShowPts[state, 0.015]

3419.1

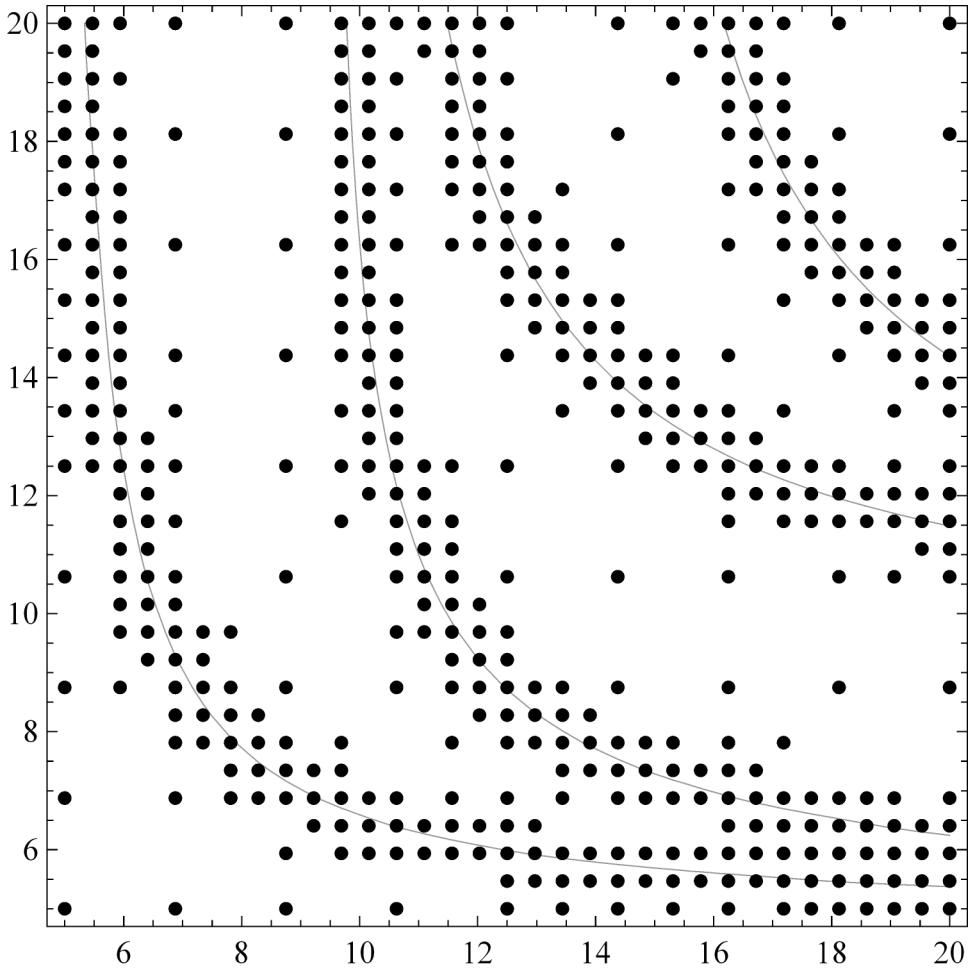
{2, 5, 25, 440}, 465 / 1089 (42.6997%)




total += Timing[state = ZLStep[state]][[1]]
ZLInfo[state]
ZLShowPts[state, 0.015]

3448.45

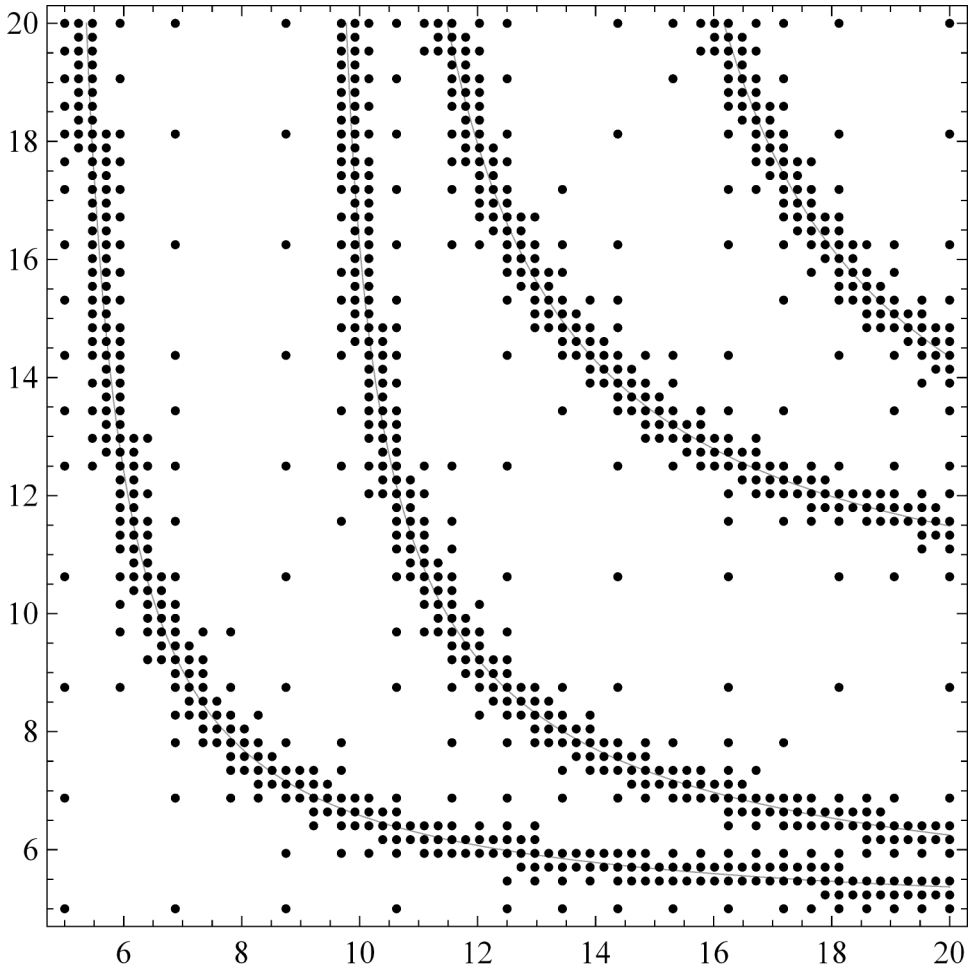
{3, 5, 81, 388}, 469 / 1089 (43.067%)




total += Timing[state = ZLStep[state]][[1]]
ZLInfo[state]
ZLShowPts[state, 0.01]

7293.97

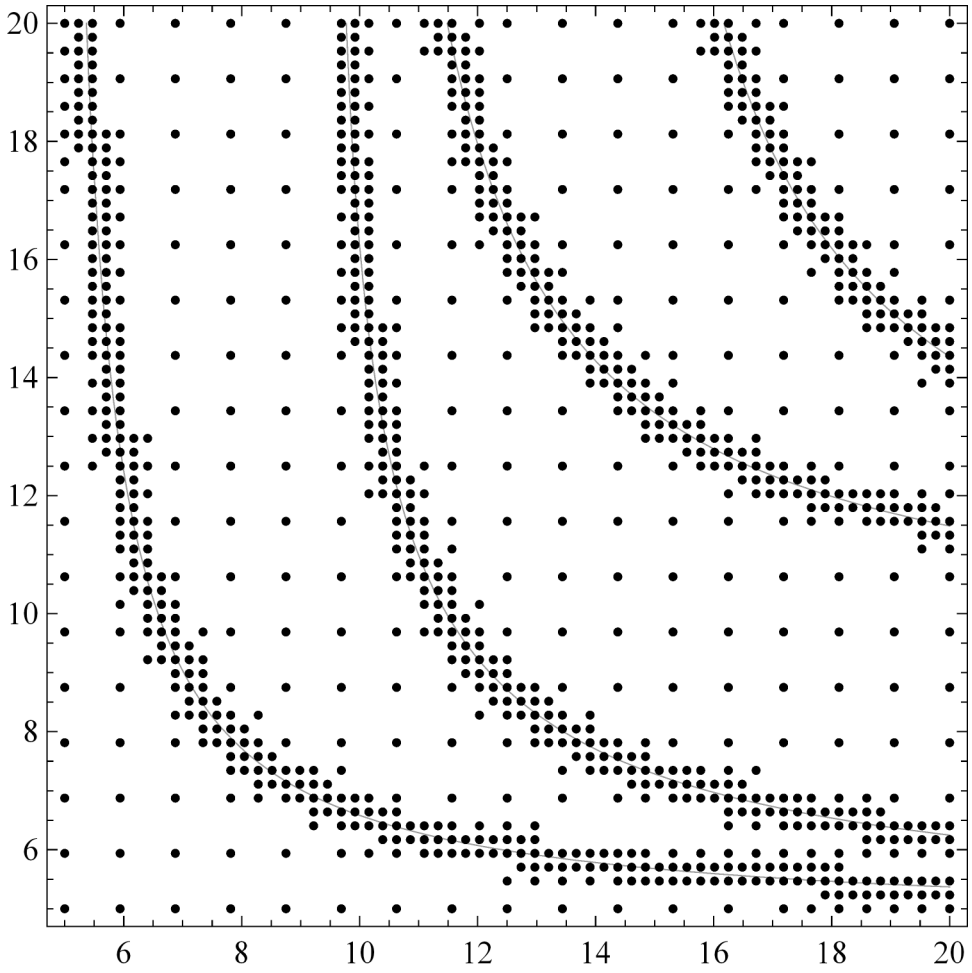
{3, 6, 81, 905}, 986 / 4225 (23.3373%)




```
total += Timing[state = ZLStep[state]][[1]]
ZLInfo[state]
ZLShowPts[state, 0.01]
```

7932.29

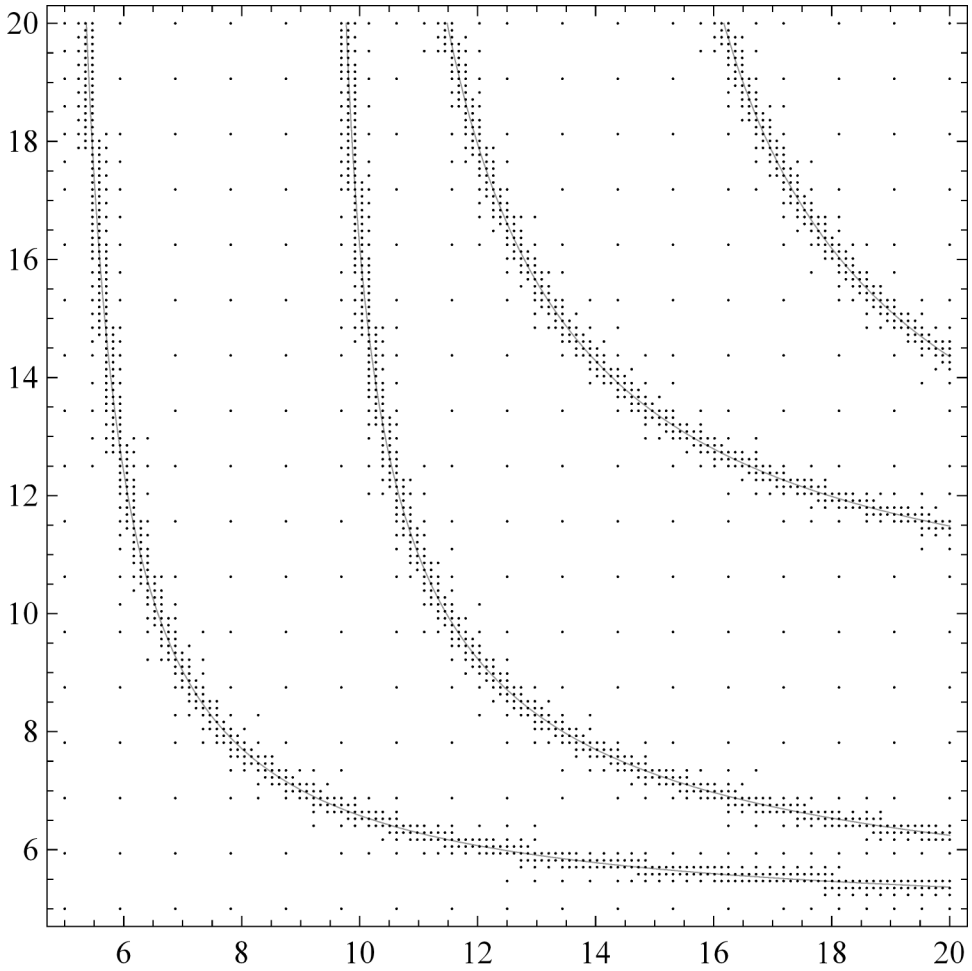
{4, 6, 289, 779}, 1068 / 4225 (25.2781%)




```
total += Timing[state = ZLStep[state]][[1]]
ZLInfo[state]
ZLShowPts[state, 0]
```

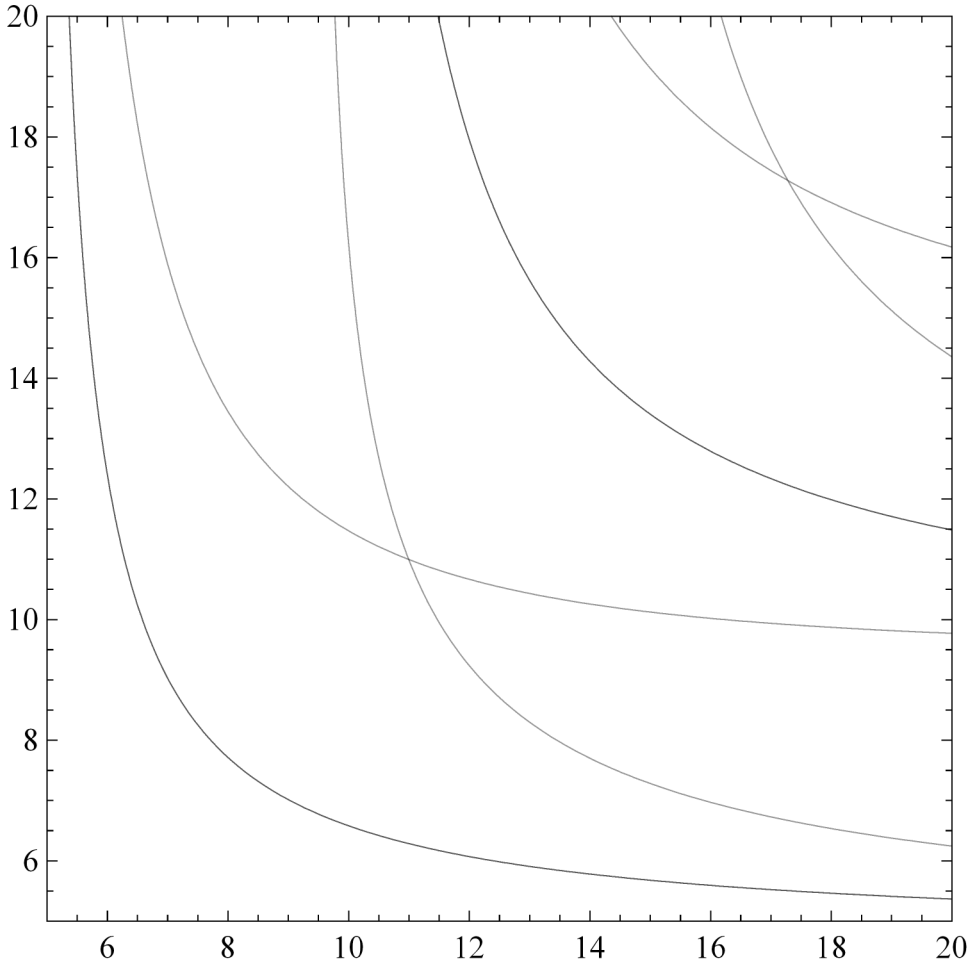
15 555.5

{4, 7, 289, 1804}, 2093 / 16641 (12.5774%)




In the last picture, the curves seem to be sufficiently smooth, so we finally display the Fučík spectru

ZLShowSym[state]




The function `ZLInfo` informs us that the function was evaluated 2093 times while the classical `Cor` hours, we can deduce that the classical `ContourPlot` would take about 1.5 day. The version of `Cc` disadvantages comparing to our algorithm - the user has to choose the value of `PlotPoints` and `M` and he realize that the curves are not smooth enough, he starts another computation without a possib

Computation on the Web

The author collaborates on web-projects based on *webMathematica* by means of which any Internet u is to collect and create eletronic materials for students adn teachers. The central portal is accesible or http://webmath.zcu.cz:8180/cmlWeb) and it is primarily designed for scientific computations. It allo e-mail with a link to the result when the computation is finished). Hence, our aim for the nearest futu

References

Benedikt, J., "Event location at integration of ODEs with jumping nonlinearity", in Applied mathem

Tajčová, G., "Mathematical models of suspension bridges", *Appl. Math.*, Vol. **42** (1997), No. 6, pp.

About the Author

The author is a mathematician working in the theory of existence, uniqueness and bifurcations of no