Reference points

In[1]:=
```
P = Import[FileNameJoin[{NotebookDirectory[], "test_system_3gen_P.txt"}], "csv"];
```

In[2]:=
```
P // MatrixForm
```

Out[2]//MatrixForm=

$$\begin{pmatrix} 24.183653333565058 & -13.5322198810934790 & -10.6514334524715770 \\ -44.3537244632853530 & 77.5839521104410890 & -33.230227647155736 \\ -78.322897115061636 & -73.917360343694043 & 152.240257458755680 \end{pmatrix}$$

In[3]:=
```
bopt = 2 * Sqrt[RankedMax[Re[Eigenvalues[P]], 2]]
```

Out[3]= 17.3795997775874592

In[4]:=
```
b1 = {bopt, bopt, bopt}
```

Out[4]= {17.3795997775874592, 17.3795997775874592, 17.3795997775874592}

In[5]:=
```
b2 = Import[FileNameJoin[{NotebookDirectory[], "betaBD0000.dat"}], "Table"][[-1]]
```

Out[5]= {17.1033, 14.9314, 24.0312}

Cusp traces

In[6]:=
```
raw1 = Import[FileNameJoin[{NotebookDirectory[], "trace0001.dat"}], "Table"];
```

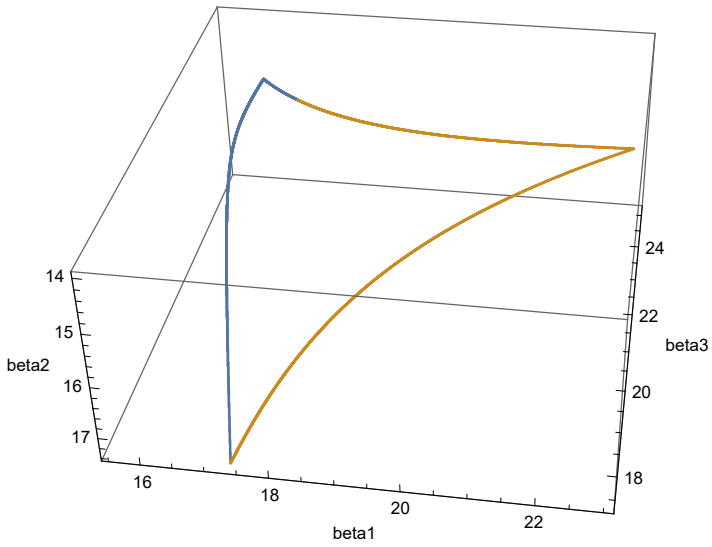
In[7]:=
```
pts1 = raw1[[2 ;;, 1 ;; 3]];
```

In[8]:=
```
raw2 = Import[FileNameJoin[{NotebookDirectory[], "trace0002.dat"}], "Table"];
```

In[9]:=
```
pts2 = raw2[[2 ;;, 1 ;; 3]];
```

In[10]:=
```
ListPointPlot3D[{pts1, pts2},
 AxesLabel → {"beta1", "beta2", "beta3"}, BoxRatios → Automatic]
```

Out[10]=



Stability function

```
In[11]:= J = ArrayFlatten[{{0 * IdentityMatrix[3], IdentityMatrix[3]}, {-P, -IdentityMatrix[3]}}];

In[12]:= J = ArrayFlatten[
            {{0 * IdentityMatrix[3], IdentityMatrix[3]}, {-P, -DiagonalMatrix[{d1, d2, d3}]}}];

In[13]:= Stab[b_] := RankedMax[Re[Eigenvalues[J /. {d1 → b[[1]], d2 → b[[2]], d3 → b[[3]]}]], 2];

In[14]:= Stab[b2]
Out[14]= -9.41029

In[15]:= dir = b2 - b1
Out[15]= {-0.27634, -2.44825, 6.65155}
```

tangent hyperplane

right eigenvector of α_2

```
In[16]:= e1 = Eigensystem[P]
Out[16]= {{178.49524079548231, 75.512622107279515, 1.4828108681026647 × 10^-15},
          {{-0.039842331747244698, -0.29668857611394253, 0.95414279717682920},
           {0.31501250831741039, -0.82351655362589497, -0.47179190911640474},
           {0.57735026918962576, 0.57735026918962577, 0.57735026918962577}}}

In[17]:= v2 = e1[[2, 2]]
Out[17]= {0.31501250831741039, -0.82351655362589497, -0.47179190911640474}
```

left eigenvector of α_2

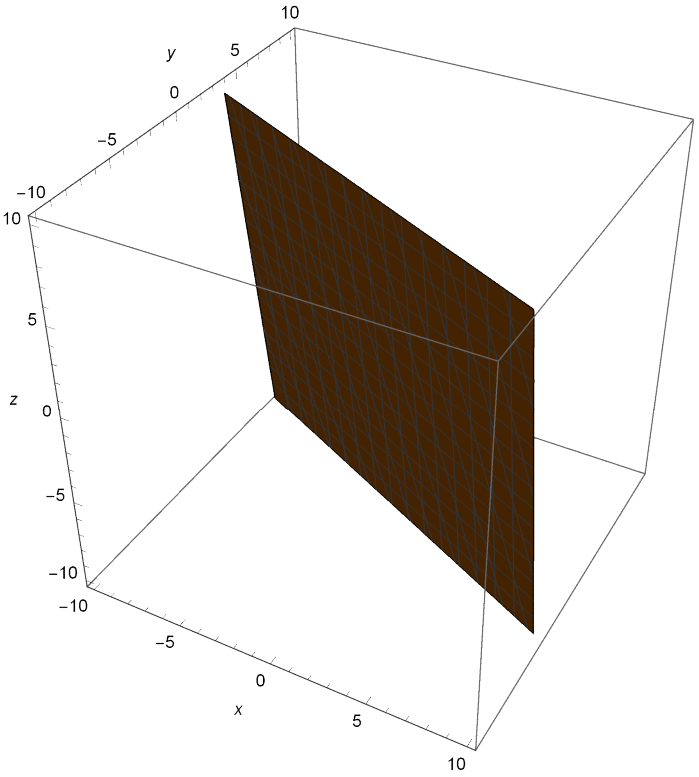
```
In[18]:= e2 = Eigensystem[Transpose[P]]
Out[18]= {{178.49524079548231, 75.512622107279515, 1.4828108681026647 × 10^-15},
          {{-0.24631568811056808, -0.55100566455591557, 0.79732135266648365},
           {0.77297931015118275, -0.61425541097109736, -0.15872389918008538},
           {0.94862376407451265, 0.28873865985700889, 0.12939451509659087}}}

In[19]:= u2 = e2[[2, 2]]
Out[19]= {0.77297931015118275, -0.61425541097109736, -0.15872389918008538}
```

hyperplane equation coefficients

```
In[20]:= L = u2 * v2
Out[20]= {0.24349815136818560, 0.50584949908897585, 0.07488465141657223}
```

In[21]:=
```
ContourPlot3D[L.{x, y, z} == 0, {x, -10, 10},
 {y, -10, 10}, {z, -10, 10}, AxesLabel → {x, y, z}]
```

Out[21]=



Express z as a function of x,y

In[22]:=
```
eqL = Solve[L.{x, y, z} == 0, z][[1, 1, 2]]
```

Out[22]=
```
13.353871335223930 (-0.24349815136818560 x - 0.50584949908897585 y)
```

The plane's coefficients are also the components of the normal vector.

In[23]:=
```
nor = Normalize[L]
```

Out[23]=
```
{0.42992247315854408, 0.8931323152655263, 0.13221699777895977}
```

Find a random point in the plane. The vector from origo to this point, normalized, is ok for one basis.

In[24]:=
```
p1 = Normalize[{1, 0, eqL /. {x → 1, y → 0}}]
```

Out[24]=
```
{0.2939500962938815, 0, -0.9558207681824129}
```

The other vector is simply the cross product of the normal and p1

In[25]:=
```
p2 = Cross[nor, p1]
```

Out[25]=
```
{-0.8536744156656323, 0.4497940277820955, -0.26253633007547875}
```

In[26]:=
```
{lb1, lb2, lb3} = Orthogonalize[{L, {1, 0, 0}, {0, 0, 1}}]
```

Out[26]=
```
{{0.42992247315854408, 0.8931323152655263, 0.13221699777895977},
 {0.9028658078990703, -0.42528762356198975, -0.06295847974462501},
 {-8.134654927793564 × 10^-40, -0.14644147183580138, 0.9892193363084671}}
```

Note: lb2 and lb3 make an orthonormal basis for the plane, more or less aligned to axes beta_1 and beta_3

plot final

In[27]:=
```
contours = Table[i, {i, -9.5, -7.5, 0.1}];
```

In[28]:=
```
colors = ColorData["LakeColors"]
```

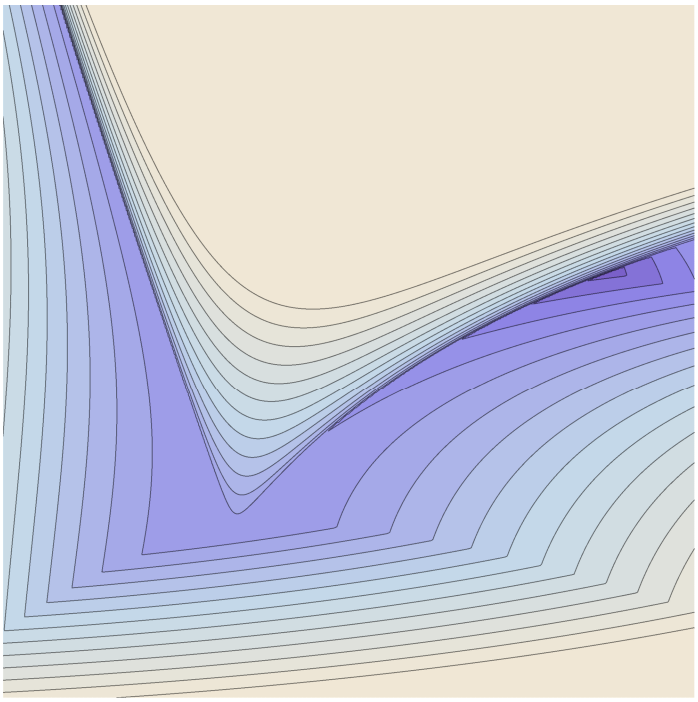
Out[28]=
```
ColorDataFunction[ Name: LakeColors
                   Gradient: ]
```

Hyperplane L:

In[29]:=
```
dataL2 = ParallelTable[Stab[b1 + x * lb2 + y * lb3], {y, -3, 9.5, 0.025}, {x, -4, 8, 0.025}];
```

In[30]:=
```
plt02 = ListContourPlot[dataL2, Contours → contours, PlotRangePadding → None,
  Frame → False, ColorFunctionScaling → False, ColorFunction → (colors[(# + 9.5) / 2] &)]
```

Out[30]=



In[31]:=
```
im02 = Rasterize[plt02, RasterSize → 600];
```

Plane crossing btilde and bg AND perpedicular to L

In[32]:=
```
basis1 = -dir;
```

In[33]:=
```
basis2 = -lb1;
```

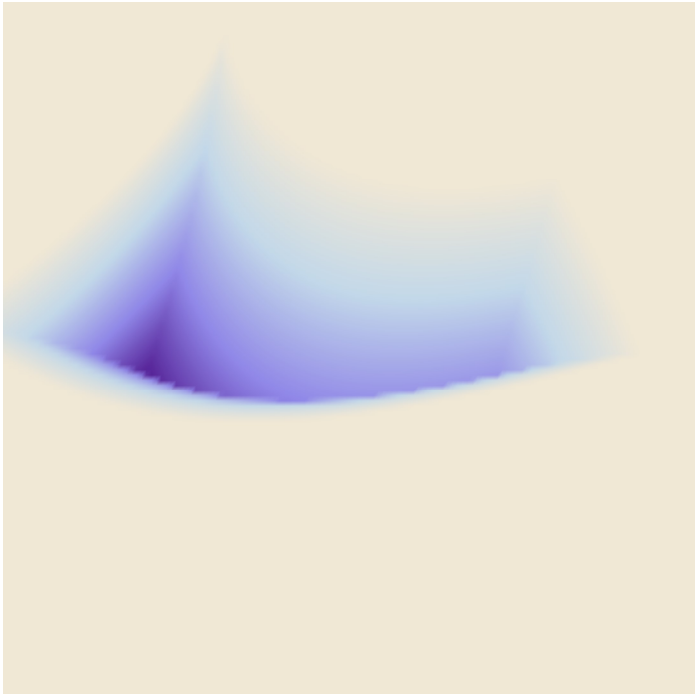
In[34]:=
```
{mb1, mb2, mb3} = Orthogonalize[{basis1, basis2, {1, 0, 0}}]
```

Out[34]=
```
{{0.0389584, 0.345155, -0.937737},
 {-0.430887, -0.840912, -0.327418}, {0.901564, -0.416815, -0.115962}}
```

In[35]:=
```
dataM = ParallelTable[Stab[bopt + x * mb1 + y * mb2], {y, -6, 7, 0.1}, {x, -10, 4, 0.1}];
```

In[36]:=
```
plt01 = ListDensityPlot[dataM, PlotRangePadding → None, Frame → False,
  ColorFunctionScaling → False, ColorFunction → (colors[(# + 9.5) / 2] &), Mesh → None]
```

Out[36]=



In[37]:=
```
im01 = Rasterize[plt01, RasterSize → 800];
```

In[38]:=
```
droplines1 = Join[
   Table[{pts1[[i]], pts1[[i]] - (pts1[[i]] - b1).lb1 * lb1}, {i, 1, 420, 100}],
   Table[{pts1[[i]], pts1[[i]] - (pts1[[i]] - b1).lb1 * lb1}, {i, 420, Length[pts1], 800}]];
```

In[39]:=
```
droplines2 =
  Table[{pts2[[i]], pts2[[i]] - (pts2[[i]] - b1).lb1 * lb1}, {i, 1, Length[pts2], 100}];
```

Planes and viewbox intersections

In[40]:=
```
eqL = lb1.{β1, β2, β3} == lb1.b1
```

Out[40]= $0.42992247315854408\ \beta_1 + 0.8931323152655263\ \beta_2 + 0.13221699777895977\ \beta_3 == 25.292041211823488$

In[41]:=
```
eqM = mb3.{β1, β2, β3} == mb3.b1
```

Out[41]= $0.901564\ \beta_1 - 0.416815\ \beta_2 - 0.115962\ \beta_3 == 6.40938$

In[42]:=
```
Solve[{eqL, eqM}, {β1, β2}][[1, 2]]
```

Out[42]= $\beta_2 \to 20.3643 - 0.171733\ \beta_3$

In[43]:=
```
label3D[s_, pos_, xVec_, tiltAngle_, opts : OptionsPattern[]] :=
  Module[{ra, width, height, r}, ra =
    Rasterize[Style[HoldForm[s], FilterRules[{opts}, Options[Style]], Magnification → 10],
     Evaluate@Apply[Sequence, FilterRules[{opts}, Options[Rasterize]]], "Image"];
   {width, height} = ImageDimensions[ra];
   r = SetAlphaChannel[ra,
     With[{color = Apply[List, ColorConvert["TransparentColor" /. {opts} /.
           {"TransparentColor" → Apply[RGBColor, ImageData[ra][[2, 2]]]}, "RGB"]]},
      Binarize[ra, (Norm[# - color] > .005) &]]];
   Translate[(* //to make lefthand corner pos*)
    Rotate[(*   //around z axis*)Rotate[(* //around y axis*)
      Rotate[(* //tilt around x axis*)Scale[(*//to make width equal|xVec|*)
        {EdgeForm[FrameStyle /. {opts} /. FrameStyle → None], Texture[ImageData@r],
         (* //Texture fills polygon initially in the xz plane*)
         Polygon[{{0, 0, 0}, {width, 0, 0}, {width, 0, height}, {0, 0, height}},
          VertexTextureCoordinates → {{0, 0}, {1, 0}, {1, 1}, {0, 1}}]},
        Norm[xVec] / width, {0, 0, 0}], tiltAngle, {1, 0, 0}], (* //x rotation*)
      Arg[Chop@N[Norm[xVec[[1 ;; 2]]] + I xVec[[3]]]], {0, -1, 0}],
     (* //y rotation*)Arg[Chop@N[xVec[[1]] + I xVec[[2]]]],
     {0, 0, 1}], (* //z rotation*)pos]];
SetAttributes[label3D, HoldFirst]
```

In[45]:=
```
imM = Rasterize[Style["M", FontFamily → "Arial", Black, Bold, 32],
  RasterSize → 200, Background → None]
```

Out[45]=

In[46]:=
```
{immx, immy} = ImageDimensions[imM] * 0.005;
```

In[47]:=
```
imL = Rasterize[Style["L", FontFamily → "Arial", Black, Bold, 32],
  RasterSize → 200, Background → None]
```

Out[47]=

In[48]:=
```
{imlx, imly} = ImageDimensions[imL] * 0.004
```

Out[48]=
```
{0.78, 1.432}
```

In[49]:=
```
{vp, vv} = Options[Graphics3D, {ViewPoint, ViewVertical}][[All, 2]];
```

Main plot

Parameters

In[50]:=
```
b1min = 14; b1max = 24; b2min = 12; b2max = 20; b3min = 16; b3max = 26;
pref = b1 - mb1 * 1.7 + mb2 * 3.9 + {0, 0.1, 0};
pref2 = b1 - lb2 * 3.6 + lb3 * -0.1 - {0, 0.01, 0};
edge1 = {β1, Solve[eqL, β2][[1, 1, 2]] /. β3 → b3max, b3max};
edge2 = {β1, Solve[eqL, β2][[1, 1, 2]] /. β3 → b3min, b3min};
edge3 = {b1max, Solve[eqL, β2][[1, 1, 2]] /. β1 → b1max, β3};
edge4 = {b1min, Solve[eqL, β2][[1, 1, 2]] /. β1 → b1min, β3};
edge5 = {Solve[eqM, β1][[1, 1, 2]] /. β2 → b2min, b2min, β3};
edge6 = {Solve[eqM, β1][[1, 1, 2]] /. β2 → b2max, b2max, β3};
edge7 = {Solve[eqM, β1][[1, 1, 2]] /. β3 → b3min, β2, b3min};
edge8 = {Solve[eqM, β1][[1, 1, 2]] /. β3 → b3max, β2, b3max};
intersect =
  {Solve[{eqL, eqM}, {β1, β2}][[1, 1, 2]], Solve[{eqL, eqM}, {β1, β2}][[1, 2, 2]], β3};
```

2D crossection (im01) and tracing the cusp on plane M

NOTE: Execute the 3D plot's configuration lines first, to get b2min, b2max, etc.

In[62]:=
```
contours2 = Table[i, {i, -9.4, -4, 0.1}];
```

In[63]:=
```
colors2 = ColorData["LakeColors"]
```

Out[63]=
```
ColorDataFunction[ Name: LakeColors
                   Gradient: ]
```

In[64]:=
```
p2 = {dir.mb1, dir.mb2}
```

Out[64]=
$$\{-7.09319,\ -4.44089 \times 10^{-16}\}$$

In[65]:=
```
planePts = Solve[eqM, β1][[1, 1, 2]]
```

Out[65]=
$$1.10918\ (6.40938 + 0.416815\ \beta_2 + 0.115962\ \beta_3)$$

In[66]:=
```
dp2 = ParallelTable[Stab[{planePts, β2, β3}],
    {β2, b2max, b2min - 1, -0.02}, {β3, b3max + 0.5, b3min - 0.5, -0.02}];
```

NOTE: the b3max+0.5 and b3min-0.5 offsets must match the datarange settings on the 2d plot

In[67]:=
```
dp2t = Transpose[dp2];
```

In[68]:=
```
minsTrace = Table[{i, Ordering[dp2t[[i]], 1][[1]]}, {i, 1, Length[dp2t]}];
```

In[69]:=
```
(b2max - b2min + 1) / 0.02
```

Out[69]=
```
450.
```

In[70]:=
```
betas2 =
  Table[b2max - Ordering[dp2t[[i]], 1][[1]] / 450 * (b2max - b2min + 1), {i, 1, Length[dp2t]}];
```

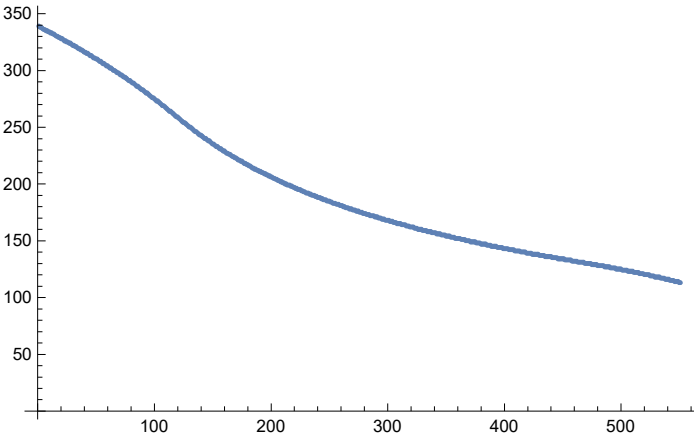
In[71]:=
```
betas3 = Table[b, {b, b3max + 0.5, b3min - 0.5, -0.02}];
```

In[72]:=
```
betas1 = Table[planePts /. {β2 → betas2[[i]], β3 → betas3[[i]]}, {i, 1, Length[betas3]}];
```

In[73]:=
```
betaTrace = Table[{betas3[[i]], betas2[[i]]}, {i, 1, Length[betas3]}];
```

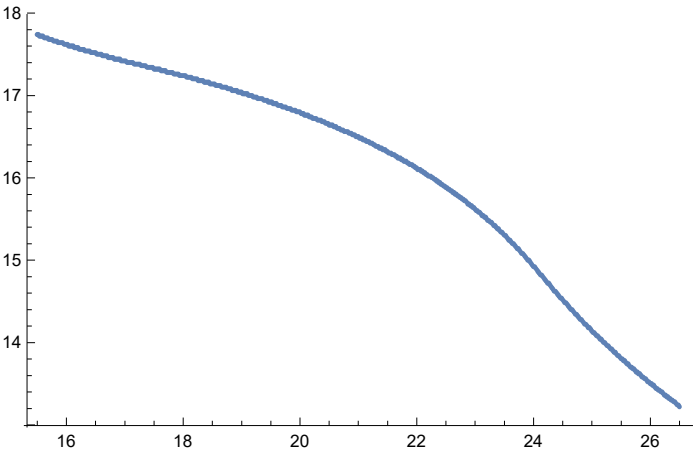
In[74]:=
```
ListPlot[minsTrace]
```

Out[74]=



In[75]:=
```
ListPlot[betaTrace]
```

Out[75]=

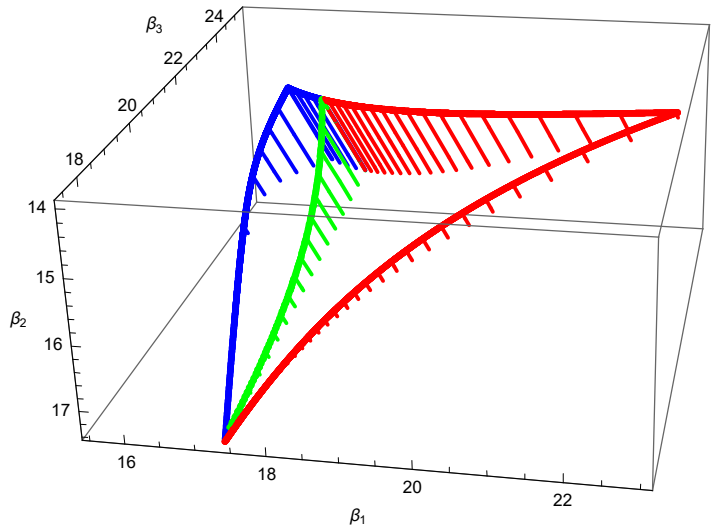


In[76]:=
```
betaFullTrace = Table[{betas1[[i]], betas2[[i]], betas3[[i]]}, {i, 125, 445}];
```

In[77]:=
```
droplines3 =
  Table[{betaFullTrace[[i]], betaFullTrace[[i]] - (betaFullTrace[[i]] - b1).lb1 * lb1},
   {i, 1, Length[betaFullTrace], 20}];
```

In[78]:=
```
Show[ListPointPlot3D[pts2, AxesLabel → {"β1 ", "β2", "β3"},
  BoxRatios → Automatic, PlotStyle → Directive[Red, PointSize[0.01]], Boxed → True],
 ListPointPlot3D[pts1, PlotStyle → Directive[Blue, PointSize[0.01]]],
 Graphics3D[{Red, Thick, Line[droplines2]}],
 Graphics3D[{Blue, Thick, Line[droplines1]}],
 Graphics3D[{Green, Thick, Line[droplines3]}],
 ListPointPlot3D[betaFullTrace, PlotStyle → Directive[Green, PointSize[0.01]]],
 PlotRange -> All, BoxRatios → Automatic]
```

Out[78]=



In[79]:=
```
pubVP = {-2.9444405070304205`, -1.0929901588743922`, -1.2592230196284422`};
```

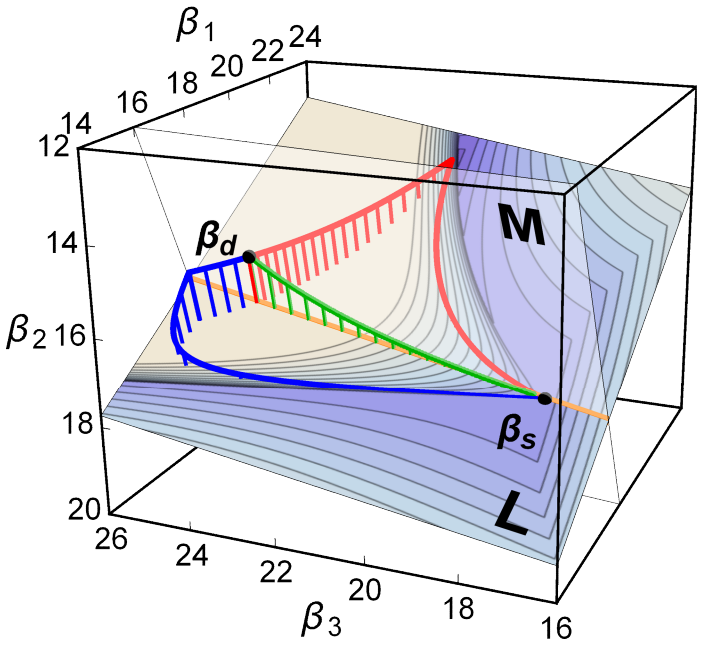
In[80]:=
```
pubVV = {-0.27037868891110106`, -1.1927941586895952`, -0.12779008326212804`};
```

In[81]:=
```
plot1 = Show[
  ListPointPlot3D[pts2, AxesLabel → {"β1 ", "β2", "β3"},
   BoxRatios → Automatic, PlotStyle → Directive[Red, PointSize[0.01]],
   Boxed → True, LabelStyle → Directive[FontFamily -> "Arial"]],
  ListPointPlot3D[pts1, PlotStyle → Directive[Blue, PointSize[0.01]]],
  ListPointPlot3D[betaFullTrace,
   PlotStyle → Directive[RGBColor[0, 0.7, 0], PointSize[0.01]]],
  ParametricPlot3D[edge1, {β1, b1min, b1max}, PlotStyle → Directive[Black, Thin]],
  ParametricPlot3D[edge2, {β1, b1min, b1max}, PlotStyle → Directive[Black, Thin]],
  ParametricPlot3D[edge3, {β3, b3min, b3max}, PlotStyle → Directive[Black, Thin]],
  ParametricPlot3D[edge4, {β3, b3min, b3max}, PlotStyle → Directive[Black, Thin]],
  ParametricPlot3D[edge5, {β3, b3min, b3max}, PlotStyle → Directive[Black, Thin]],
  ParametricPlot3D[edge6, {β3, b3min, b3max}, PlotStyle → Directive[Black, Thin]],
  ParametricPlot3D[edge7, {β2, b2min, b2max}, PlotStyle → Directive[Black, Thin]],
  ParametricPlot3D[edge8, {β2, b2min, b2max}, PlotStyle → Directive[Black, Thin]],
  ParametricPlot3D[intersect, {β3, b3min, b3max},
   PlotStyle → Directive[Orange, Thickness[0.015]]],
  Graphics3D[{Red, Thick, Line[droplines2]}],
  Graphics3D[{Blue, Thick, Line[droplines1]}],
  Graphics3D[{RGBColor[0, 0.7, 0], Thick, Line[droplines3]}],
```

```
Graphics3D[{
  Texture[im02],
  Lighting → {{"Ambient", White}},
  Polygon[{
    b1 + lb2 * 8 + lb3 * 9.5,
    b1 + lb2 * 8 – lb3 * 3,
    b1 – lb2 * 4 – lb3 * 3,
    b1 – lb2 * 4 + lb3 * 9.5
   }, VertexTextureCoordinates → {{1, 1}, {1, 0}, {0, 0}, {0, 1}}]}],
Graphics3D[{Lighting → {{"Ambient", Black}}, Sphere[b1, 0.15]}],
Graphics3D[{Lighting → {{"Ambient", Black}}, Sphere[b2, 0.15]}],
Graphics3D[{Text[
   Style["βs", FontSize → 21, FontWeight → Bold, FontColor → Black], b1, {1.4, 1.2}]}],
Graphics3D[{Text[Style["βd", FontSize → 21, FontWeight → Bold, FontColor → Black],
   b2, {1.5, –1}]}],
Graphics3D[{
  Texture[imM],
  Lighting → {{"Ambient", White}},
  EdgeForm[None],
  Rotate[
   Polygon[{
     pref + mb1 * 0 + mb2 * immy,
     pref + mb1 * 0 + mb2 * 0,
     pref + mb1 * immx + mb2 * 0,
     pref + mb1 * immx + mb2 * immy
    },
    VertexTextureCoordinates → {{1, 1}, {1, 0}, {0, 0}, {0, 1}}
   ], –0.35, mb3, pref]
 }],
Graphics3D[{
  Texture[imL],
  Lighting → {{"Ambient", White}},
  EdgeForm[None],
  Polygon[{
    pref2 – lb3 * imlx + lb2 * imly,
    pref2 – lb3 * imlx + lb2 * 0,
    pref2 + lb3 * 0 + lb2 * 0,
    pref2 + lb3 * 0 + lb2 * imly},
   VertexTextureCoordinates → {{1, 1}, {1, 0}, {0, 0}, {0, 1}}]
 }],
Graphics3D[{
  Lighting → {{"Ambient", White}},
  FaceForm[White],
  Opacity[0.4],
  Polygon[{
    b1 – mb1 * 10 + mb2 * 7,
```

```
      b1 - mb1 * 10 - mb2 * 6,
      b1 + mb1 * 4 - mb2 * 6,
      b1 + mb1 * 4 + mb2 * 7
     }
    ]}],
  PlotRange → {{b1min, b1max}, {b2min, b2max}, {b3min, b3max}},
  AxesStyle → Directive[Black, 20], BoxStyle → Black,
  Ticks → {Table[i, {i, 12, 26, 2}], Table[i, {i, 12, 26, 2}], Table[i, {i, 12, 26, 2}]},
  TicksStyle → Directive[16],
  ViewPoint → pubVP, ViewVertical → pubVV, PlotRangePadding -> None,
  RenderingOptions → {"3DRenderingEngine" → "Mesa", "LegacyAlphaChannelMethod" → False}
 ]
```

Out[81]=



In[82]:= **vp**

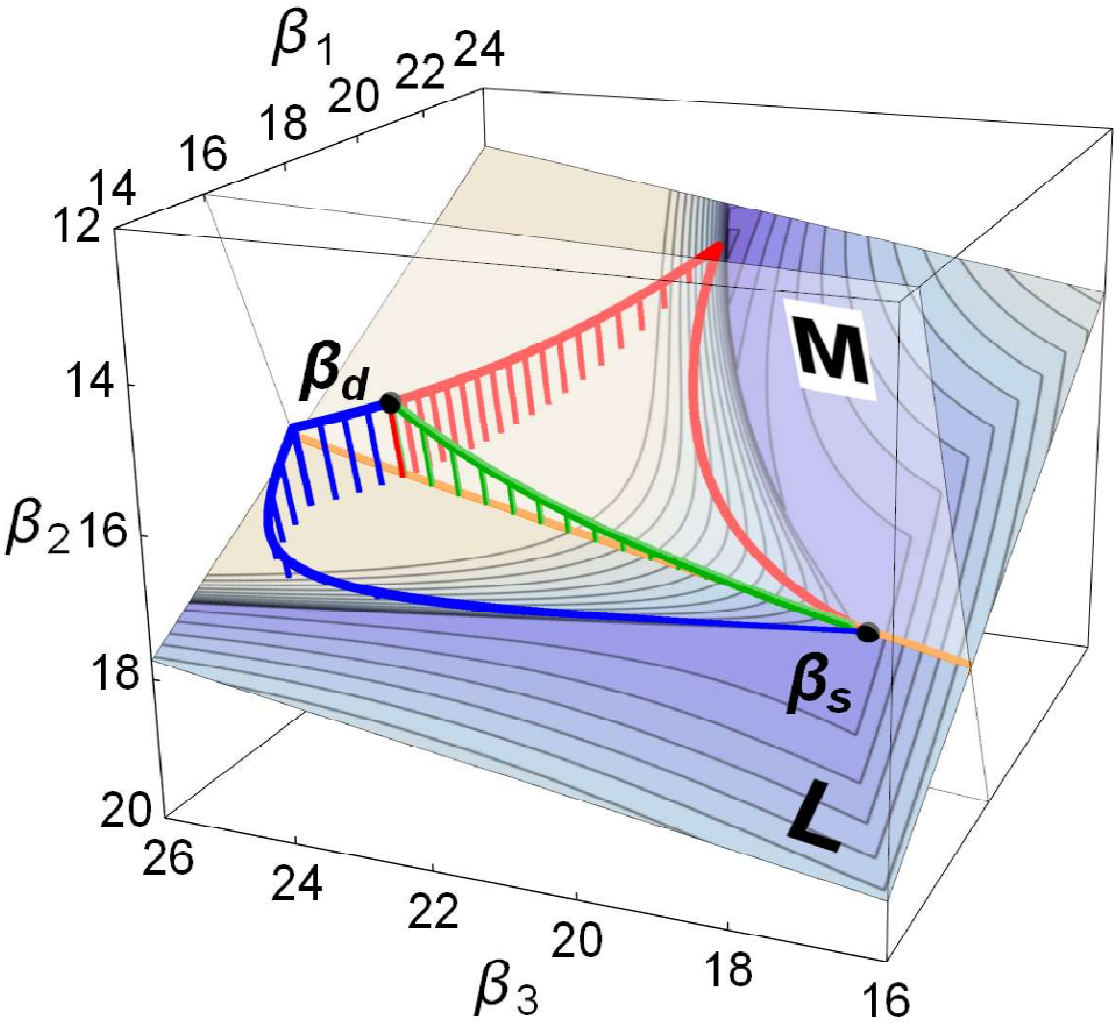
Out[82]= {1.3, -2.4, 2.}

In[83]:= **vv**

Out[83]= {0, 0, 1}

In[84]:=
```
ras1 = Rasterize[plot1, RasterSize → 1000]
```

Out[84]=



In[85]:=
```
Export[FileNameJoin[{NotebookDirectory[], "panel3D.png"}], ras1];
```

NOTE: export the original plot to PDF or EPS to avoid the white box around “M”

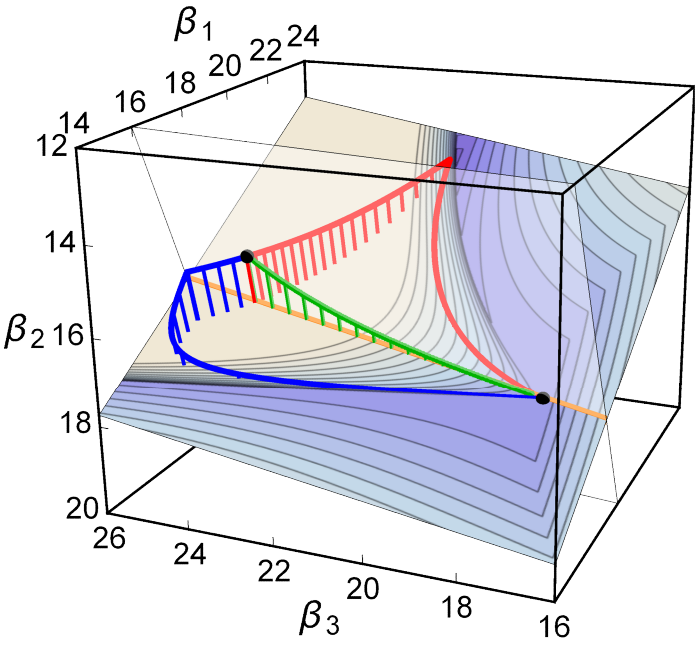
In[86]:=
```
Export[FileNameJoin[{NotebookDirectory[], "panel3D.eps"}],
  plot1, RasterSize → 1000, Antialiasing -> True];
```

BLANK version

In[87]:=
```
plot1blank = Show[
  ListPointPlot3D[pts2, AxesLabel → {"β1 ", "β2", "β3"},
   BoxRatios → Automatic, PlotStyle → Directive[Red, PointSize[0.01]],
   Boxed → True, LabelStyle → Directive[FontFamily -> "Arial"]],
  ListPointPlot3D[pts1, PlotStyle → Directive[Blue, PointSize[0.01]]],
  ListPointPlot3D[betaFullTrace,
   PlotStyle → Directive[RGBColor[0, 0.7, 0], PointSize[0.01]]],
  ParametricPlot3D[edge1, {β1, b1min, b1max}, PlotStyle → Directive[Black, Thin]],
```

```
 ParametricPlot3D[edge2, {β1, b1min, b1max}, PlotStyle → Directive[Black, Thin]],
 ParametricPlot3D[edge3, {β3, b3min, b3max}, PlotStyle → Directive[Black, Thin]],
 ParametricPlot3D[edge4, {β3, b3min, b3max}, PlotStyle → Directive[Black, Thin]],
 ParametricPlot3D[edge5, {β3, b3min, b3max}, PlotStyle → Directive[Black, Thin]],
 ParametricPlot3D[edge6, {β3, b3min, b3max}, PlotStyle → Directive[Black, Thin]],
 ParametricPlot3D[edge7, {β2, b2min, b2max}, PlotStyle → Directive[Black, Thin]],
 ParametricPlot3D[edge8, {β2, b2min, b2max}, PlotStyle → Directive[Black, Thin]],
 ParametricPlot3D[intersect, {β3, b3min, b3max},
  PlotStyle → Directive[Orange, Thickness[0.015]]],
 Graphics3D[{Red, Thick, Line[droplines2]}],
 Graphics3D[{Blue, Thick, Line[droplines1]}],
 Graphics3D[{RGBColor[0, 0.7, 0], Thick, Line[droplines3]}],
 Graphics3D[{
   Texture[im02],
   Lighting → {{"Ambient", White}},
   Polygon[{
     b1 + lb2 * 8 + lb3 * 9.5,
     b1 + lb2 * 8 – lb3 * 3,
     b1 – lb2 * 4 – lb3 * 3,
     b1 – lb2 * 4 + lb3 * 9.5
    }, VertexTextureCoordinates → {{1, 1}, {1, 0}, {0, 0}, {0, 1}}]}],
 Graphics3D[{Lighting → {{"Ambient", Black}}, Sphere[b1, 0.15]}],
 Graphics3D[{Lighting → {{"Ambient", Black}}, Sphere[b2, 0.15]}],
 Graphics3D[{
   Lighting → {{"Ambient", White}},
   FaceForm[White],
   Opacity[0.4],
   Polygon[{
     b1 – mb1 * 10 + mb2 * 7,
     b1 – mb1 * 10 – mb2 * 6,
     b1 + mb1 * 4 – mb2 * 6,
     b1 + mb1 * 4 + mb2 * 7
    }
   ]}],
 PlotRange → {{b1min, b1max}, {b2min, b2max}, {b3min, b3max}},
 AxesStyle → Directive[Black, 20], BoxStyle → Black,
 Ticks → {Table[i, {i, 12, 26, 2}], Table[i, {i, 12, 26, 2}], Table[i, {i, 12, 26, 2}]},
 TicksStyle → Directive[16],
 ViewPoint → pubVP, ViewVertical → pubVV, PlotRangePadding -> None,
 RenderingOptions → {"3DRenderingEngine" → "Mesa", "LegacyAlphaChannelMethod" → False}
]
```

Out[87]=



In[88]:=
```
Export[FileNameJoin[{NotebookDirectory[], "panel3D_blank.eps"}],
  plot1blank, RasterSize → 1000, Antialiasing -> True];
```

corner coordinates of plane M in beta-space

In[89]:=
```
hticks = Table[{i, 42 - i}, {i, b3min, b3max, 2}]
```

Out[89]= {{16, 26}, {18, 24}, {20, 22}, {22, 20}, {24, 18}, {26, 16}}

In[90]:=
```
vticks = Table[{i, 31 - i}, {i, b2min - 1, b2max, 2}]
```

Out[90]= {{11, 20}, {13, 18}, {15, 16}, {17, 14}, {19, 12}}

In[91]:=
```
p1 = {42 - bopt, 31 - bopt};
```

In[92]:=
```
p2 = {42 - b2[[3]], 31 - b2[[2]]};
```

In[93]:=
```
isect = Solve[{eqL, eqM}, {β2, β3}][[1]]
```

Out[93]= $\{\beta_2 \to 78.0103 - 3.48862\,\beta_1,\ \beta_3 \to -335.672 + 20.3141\,\beta_1\}$

In[94]:=
```
isectPt = {42 - isect[[2, 2]], 31 - isect[[1, 2]]}
```

Out[94]= $\{377.672 - 20.3141\,\beta_1,\ -47.0103 + 3.48862\,\beta_1\}$

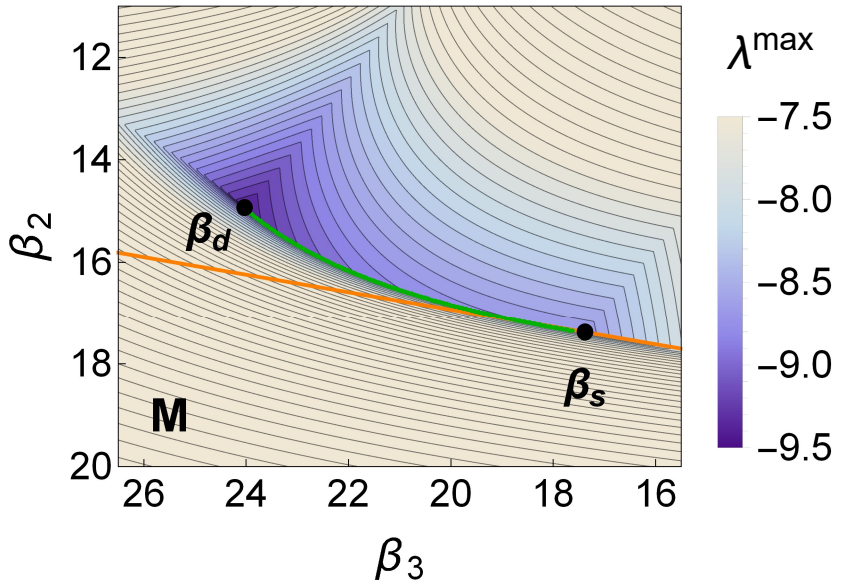
In[95]:=
```
isectPt /. β1 → bopt
```

Out[95]= {24.6204, 13.6204}

In[96]:=
```
betaTrace2 = Table[{42 - betas3[[i]], 31 - betas2[[i]] - 0.05}, {i, 125, 445}];
```

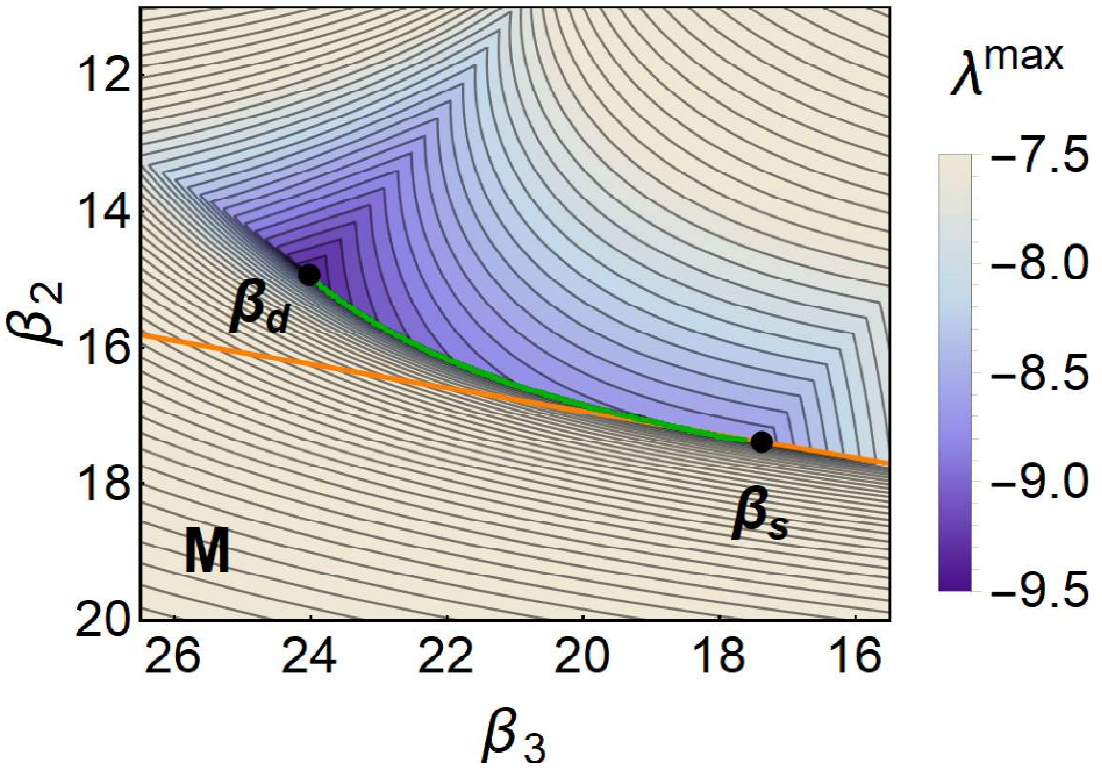
In[97]:=
```
cp1 = ListContourPlot[dp2, DataRange → {{b3min - 0.5, b3max + 0.5}, {b2min - 1, b2max}},
    FrameTicks → {{vticks, None}, {hticks, None}},
    PlotRangePadding → None, Contours → contours2, ColorFunctionScaling → False,
    ColorFunction → (colors2[Rescale[#, {-9.5, -7.5}]] &),
    FrameLabel → {"β3", "β2"}, FrameStyle → Directive[Black, 24],
    FrameTicksStyle → Directive[20], AspectRatio → Automatic];
plot3 = Show[
  Legended[cp1, BarLegend[{"LakeColors", {-9.5, -7.5}},
    LegendLabel → Style[λ^max, FontSize → 24], LegendMargins → {{0, 0}, {0, 0}},
    LegendMarkerSize → 200, LabelStyle → Directive[20, Black]]],
  ParametricPlot[isectPt, {β1, 17, 18}, PlotStyle → Directive[Orange, Thick]],
  ListPlot[betaTrace2, PlotStyle → Directive[RGBColor[0, 0.7, 0], Thick]],
  Graphics[{PointSize[0.03], Black, Point[p1]}],
  Graphics[{PointSize[0.03], Black, Point[p2]}],
  Graphics[
   Text[Style[βs, FontSize → 22, FontWeight → Bold, FontColor → Black], p1 + {0, -1}]],
  Graphics[Text[Style[βd, FontSize → 22, FontWeight → Bold, FontColor → Black],
    p2 + {-0.7, -0.4}]],
  Graphics[Text[Style["M", FontSize → 24, FontWeight → Bold, FontColor → Black],
    {16.5, 12}]],
  ImagePadding → {{62, 5}, {62, 40}}
 ]
```

Out[98]=



In[99]:=
```
ras2 = Rasterize[plot3, RasterSize → 1000]
```

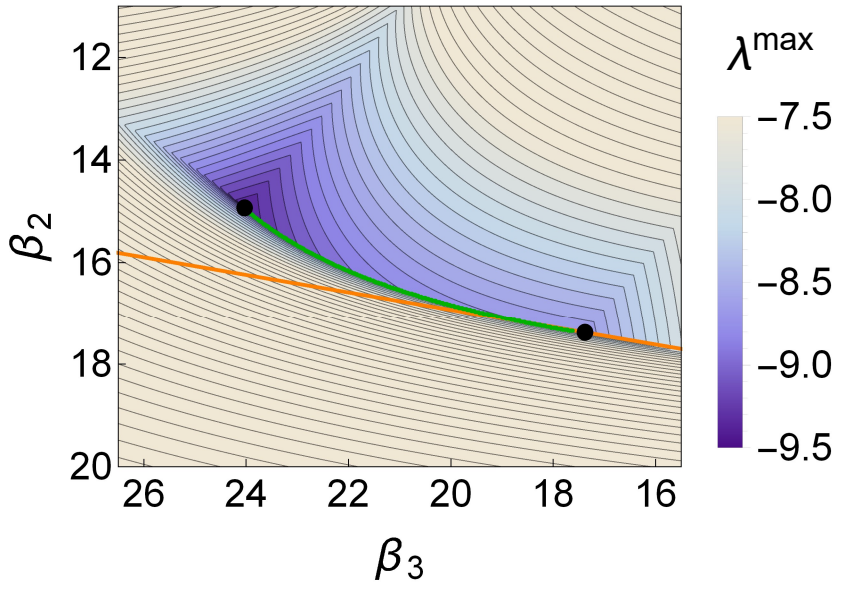
Out[99]=



In[100]:=
```
Export[FileNameJoin[{NotebookDirectory[], "panel2D.eps"}], ras2];
```

In[101]:=
```
cp1 = ListContourPlot[dp2, DataRange → {{b3min - 0.5, b3max + 0.5}, {b2min - 1, b2max}},
    FrameTicks → {{vticks, None}, {hticks, None}},
    PlotRangePadding → None, Contours → contours2, ColorFunctionScaling → False,
    ColorFunction → (colors2[Rescale[#, {-9.5, -7.5}]] &),
    FrameLabel → {"β3", "β2"}, FrameStyle → Directive[Black, 24],
    FrameTicksStyle → Directive[20], AspectRatio → Automatic];
plot3blank = Show[
  Legended[cp1, BarLegend[{"LakeColors", {-9.5, -7.5}},
    LegendLabel → Style[λ^max, FontSize → 24], LegendMargins → {{0, 0}, {0, 0}},
    LegendMarkerSize → 200, LabelStyle → Directive[20, Black]]],
  ParametricPlot[isectPt, {β1, 17, 18}, PlotStyle → Directive[Orange, Thick]],
  ListPlot[betaTrace2, PlotStyle → Directive[RGBColor[0, 0.7, 0], Thick]],
  Graphics[{PointSize[0.03], Black, Point[p1]}],
  Graphics[{PointSize[0.03], Black, Point[p2]}],
  ImagePadding → {{62, 5}, {62, 40}}
 ]
```

Out[102]=



In[103]:=
```
Export[FileNameJoin[{NotebookDirectory[], "panel2D_blank.eps"}],
  Rasterize[plot3blank, RasterSize → 1000]];
```