```
(*f[x_,y_]:=Sin[x-π/2];*)
g[x_, y_] := 1/(2*π*σ^2) * e^(-(x^2+y^2)/(2*σ^2));
f[x_, y_] := (g[x, y] + Sum[g[x + i, y], {i, 1, 10}] + Sum[g[x, y + i], {i, 1, 10}] + g[x + 10, y + 10]) /. {σ → 1};
f2[x_, y_] := Cos[x];
s[x_, y_] := Sqrt[1 - x^2 - y^2];
Hessian2D[f_] := ( D[D[f[x, y], {x}], {x}]  D[D[f[x, y], {x}], {y}]
                   D[D[f[x, y], {y}], {x}]  D[D[f[x, y], {y}], {y}] );
Hessian_f[x_, y_] = Hessian2D[f];
Hessian_f2[x_, y_] = Hessian2D[f2];
xLeft = -12;
xRight = 5;
yLeft = -12;
yRight = 5;

Hessian_f2[x, y] // MatrixForm
```

$$\begin{pmatrix} -\mathrm{Cos}[x] & 0 \\ 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} \mathrm{Cos}[45\,°] \\ \mathrm{Sin}[45\,°] \end{pmatrix}^{T}.\mathrm{Hessian}_{f2}[x, y].\begin{pmatrix} \mathrm{Cos}[45\,°] \\ \mathrm{Sin}[45\,°] \end{pmatrix}$$

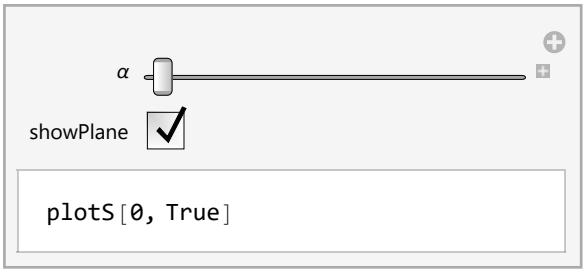
$$\left\{\left\{-\frac{\mathrm{Cos}[x]}{2}\right\}\right\}$$

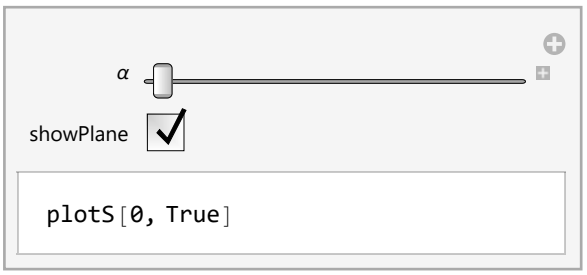
$$\mathrm{Cos}\left[\frac{\pi}{4}\right]$$

$$\frac{1}{\sqrt{2}}$$

```
plotS[α_, showPlane_?BooleanQ] :=
 Block[{function, H, s, sHs, posSlider, pos, grad, λ, e, scale, colorE, colorS, planeScale},
  function = f2;
  posSlider = {0, 0};
  pos = Append[posSlider, function[posSlider[[1]], posSlider[[2]]]];
  H = Hessian_f2[pos[[1]], pos[[2]]];
  s = (Cos[α]
       Sin[α]);
  grad = Grad[function[x, y], {x, y}] /. {x → pos[[1]], y → pos[[2]]};
  sHs = s^T.H.s;
  sHs = sHs[[1, 1]];
  λ_1 = Eigenvalues[H][[1]];
  e_1 = Normalize[Eigenvectors[H][[1]]];
  scale = 0.5 + 0.5 * { sHs/λ_1   λ_1 ≠ 0
                        0         λ_1 == 0 ;
  colorE_1 = ■;
  colorS = ■;
  planeScale = 10;

  Show[
   Plot3D[function[x, y], {x, -2, 2}, {y, -2, 2},
    ImageSize → Large,
    AxesLabel → Automatic,
    PlotStyle → Opacity[0.5],
    AspectRatio → Automatic,
    PlotLabel → Style["s⃗^T H s⃗ = " <> ToString[sHs // N], colorS],
    PlotLegends → {"f(x,y) = cos(x)"},
    Mesh → None,
    BaseStyle → {FontSize → 14}
   ],
```

```
    (* Inspiration from http://demonstrations.wolfram.com/DirectionalDerivativesIn3D/ *)
    ParametricPlot3D[{s[[1, 1]] * t + posSlider[[1]], s[[2, 1]] * t + posSlider[[2]],
      function[s[[1, 1]] * t + posSlider[[1]], s[[2, 1]] * t + posSlider[[2]]]},
     {t, -3, 3}, PlotStyle → Directive[colorS, Opacity[0.5]]],

    If[showPlane,
     Graphics3D[{
       colorS, Opacity[0.3],
       Polygon[{{-planeScale s[[1, 1]] + posSlider[[1]], -planeScale s[[2, 1]] + posSlider[[2]],
           -planeScale}, {planeScale s[[1, 1]] + posSlider[[1]],
           planeScale s[[2, 1]] + posSlider[[2]], -planeScale},
          {planeScale s[[1, 1]] + posSlider[[1]], planeScale s[[2, 1]] + posSlider[[2]], planeScale},
          {-planeScale s[[1, 1]] + posSlider[[1]],
           -planeScale s[[2, 1]] + posSlider[[2]], planeScale}}]
      }]
     ,
     {}
    ],

    Legended[
     Graphics3D[{colorE1, Thick, Opacity[0.5],
       Arrowheads[0.045], Arrow[Tube[{pos, pos + {e1[[1]], e1[[2]], 0}}, 0.02]]}],
     SwatchLegend[{colorE1, colorS}, {"ê1", "s⃗"}, LegendMarkers → {
        Graphics3D[
         {Arrowheads[0.3], Arrow[Tube[{{0, 0, 0}, {0.1, 0.1, 0.1}}, 0.005]]}, Boxed → False
        ]
       },
      LegendMarkerSize → 60
     ]
    ],

    Graphics3D[{colorS, Thick, Arrowheads[0.045],
      Arrow[Tube[{pos, pos + {s[[1, 1]], s[[2, 1]], 0} * scale}, 0.02]]}]
   ]
  ]

Manipulate[
 plotS[α, showPlane]
 , {α, 0, 2 π}, {showPlane, {True, False}}]
```






```
(*Export[FileNameJoin[{NotebookDirectory[],"frames/showPlane=0alpha=00.png"}],
 Table[plotS[α,False],{α,0,2π,π/12}],"VideoFrames",Antialiasing→True];
Export[FileNameJoin[{NotebookDirectory[],"frames/showPlane=1alpha=00.png"}],
 Table[plotS[α,True],{α,0,2π,π/12}],"VideoFrames",Antialiasing→True];*)
```

Find the maximum eigenvalue (needed for arrow scaling)

```
max1 = FindMaximum[Det[Hessian2D[f]] // Evaluate, {{x, 0}, {y, 0}}][[2, ;; , 2]]
max2 = FindMaximum[Det[Hessian2D[f]] // Evaluate, {{x, 0}, {y, -10}}][[2, ;; , 2]]
max3 = FindMaximum[Det[Hessian2D[f]] // Evaluate, {{x, -10}, {y, 0}}][[2, ;; , 2]]
max4 = FindMaximum[Det[Hessian2D[f]] // Evaluate, {{x, -10}, {y, -10}}][[2, ;; , 2]]
```

{-0.240217, -0.240217}

$\{1.73267 \times 10^{-12}, -9.42336\}$

$\{-9.42336, 1.44454 \times 10^{-11}\}$

FindMaximum: Encountered a gradient that is effectively zero. The result returned may not be a maximum; it may be a minimum or a saddle point.

{-10., -10.}

```
e1 = Eigenvalues[Hessian2D[f] /. {x → max1[[1]], y → max1[[2]]}]
e2 = Eigenvalues[Hessian2D[f] /. {x → max2[[1]], y → max2[[2]]}]
e3 = Eigenvalues[Hessian2D[f] /. {x → max3[[1]], y → max3[[2]]}]
e4 = Eigenvalues[Hessian2D[f] /. {x → max4[[1]], y → max4[[2]]}]
```

{-0.306857, -0.174445}

{-0.346992, -0.103916}

{-0.346992, -0.103916}

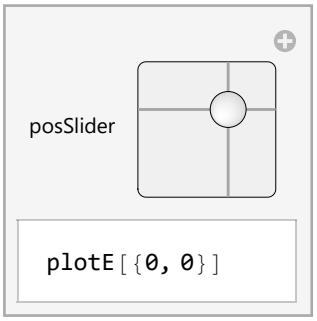
{-0.159155, -0.159155}

```
maxλ = Max[Abs[e1], Abs[e2], Abs[e3], Abs[e4]]
```

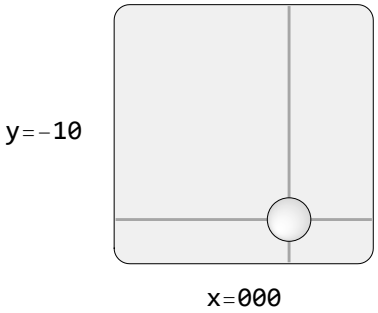
0.346992

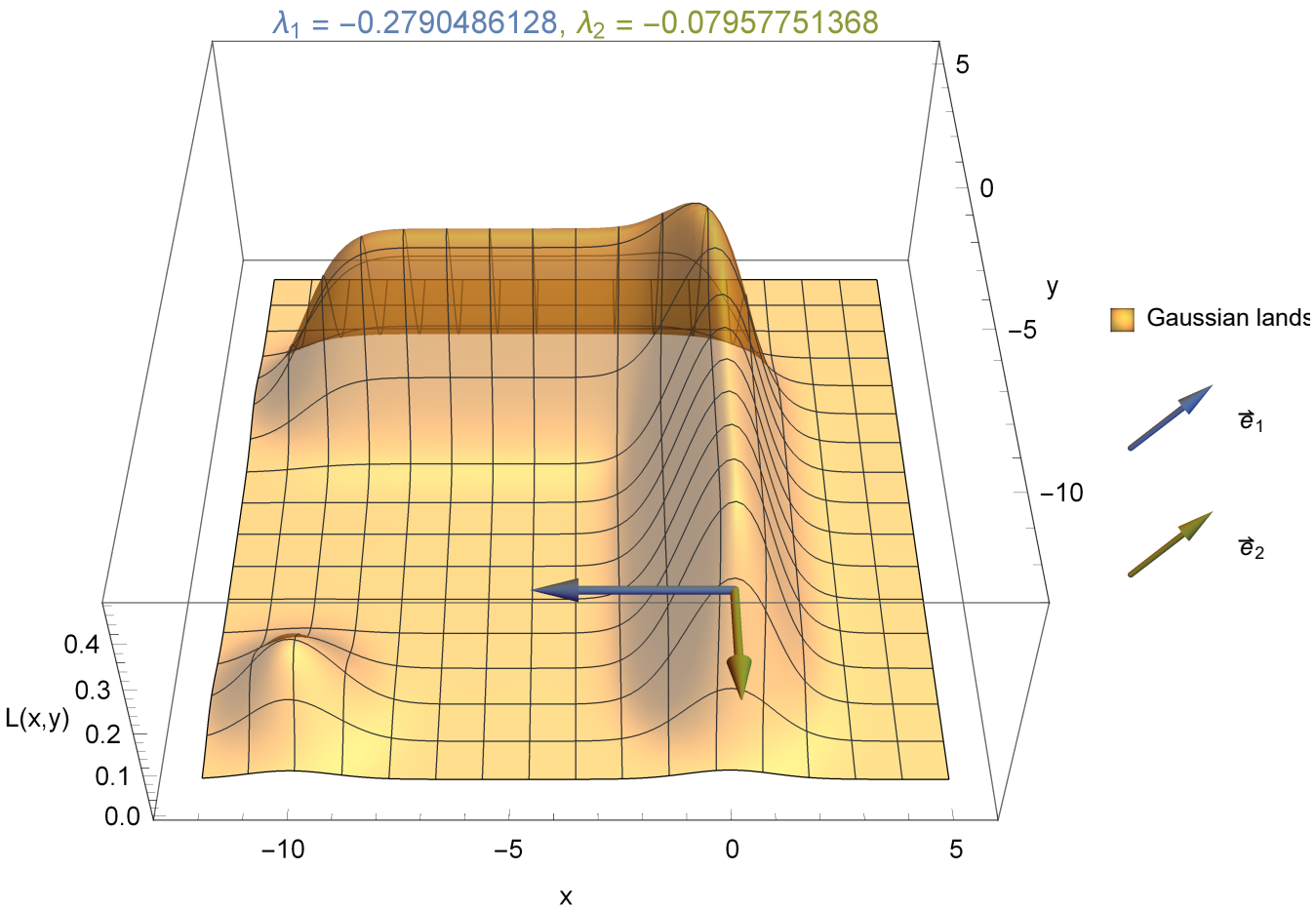
```
plotE[posSlider_] := Block[{H, e, λ, pos, grad, scaleλ, arrowLength, tubeSize, colorE},
  pos = Append[posSlider, f[posSlider[[1]], posSlider[[2]]]];
  (* It is a bit strange but when the Hessian is not set to
   numeric the eigenvectors may not be orthogonal after normalization *)
  H = Hessian_f[pos[[1]], pos[[2]]] // N;
  e_1 = Eigenvectors[H][[1]];
  e_2 = Eigenvectors[H][[2]];
  λ_1 = Eigenvalues[H][[1]];
  λ_2 = Eigenvalues[H][[2]];
  (* The scale values give a value in the range [0.5;1] *)
  scaleλ_1 = 0.5 + 0.5 * { Abs[λ_1]/maxλ   λ_1 ≠ 0
                           0               λ_1 == 0 ;
  scaleλ_2 = 0.5 + 0.5 * { Abs[λ_2]/maxλ   λ_2 ≠ 0
                           0               λ_2 == 0 ;
  arrowLength = 5;
  tubeSize = 0.1;
  colorE_1 = ■;
  colorE_2 = ■;

  Show[
   plotObj = Plot3D[f[x, y], {x, xLeft, xRight}, {y, yLeft, yRight},
     (* Make a bit extra space for the arrows at the border *)
     PlotRange → {{xLeft - 1, xRight + 1}, {yLeft - 1, yRight + 1}, Automatic},
     AxesLabel → {"x", "y", "L(x,y)"},
     PlotStyle → Opacity[0.5],
     AspectRatio → Automatic,
     PlotLabel → Row[{
        (* It is important that the scientific notation is not used since this
         would result in a different height which in turn breaks the animation *)
        Style["λ_1 = " <> ToString[NumberForm[N[λ_1, 10], 10, ExponentFunction → (Null &)],
           StandardForm], colorE_1],
```

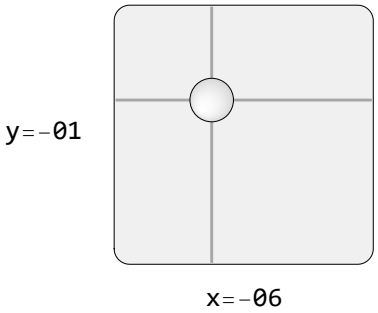
```
        Style[", λ2 = " <> ToString[NumberForm[N[λ2, 10], 10, ExponentFunction → (Null &)],
           StandardForm], colorE2]
       }],
     PlotLegends → {"Gaussian landscape"},
     ViewPoint → {0, -2, 2},
     PlotPoints → 100,
     (* More plot points result in higher quality but also slower performance *)
     BaseStyle → {FontSize → 14}
    ],

   Legended[
    Graphics3D[
     Scale[
      {colorE1, Arrow[Tube[{pos, pos + {e1[[1]], e1[[2]], 0} * scaleλ1 * arrowLength}, tubeSize]]},
      # / Max[#] &[-Subtract @@@ #[[1]] / #[[2]] &[
        Values /@ AbsoluteOptions[plotObj, {PlotRange, BoxRatios}]]]
     ]
    ],
    SwatchLegend[{colorE1, colorE2}, {"ė1", "ė2"}, LegendMarkers → {
       Graphics3D[
        {Arrowheads[0.3], Arrow[Tube[{{0, 0, 0}, {0.1, 0.1, 0.1}}, 0.005]]}, Boxed → False
       ]
      },
     LegendMarkerSize → 60
    ]
   ],

   Graphics3D[
    Scale[
     {colorE2, Arrow[Tube[{pos, pos + {e2[[1]], e2[[2]], 0} * scaleλ2 * arrowLength}, tubeSize]]},
     # / Max[#] &[-Subtract @@@ #[[1]] / #[[2]] &[
       Values /@ AbsoluteOptions[plotObj, {PlotRange, BoxRatios}]]]
    ]
   ],

   ImageSize → Large
  ]
 ]

Manipulate[
 plotE[posSlider]
 , {{posSlider, {0, 0}}, {xLeft, yLeft}, {xRight, yRight}}]
```

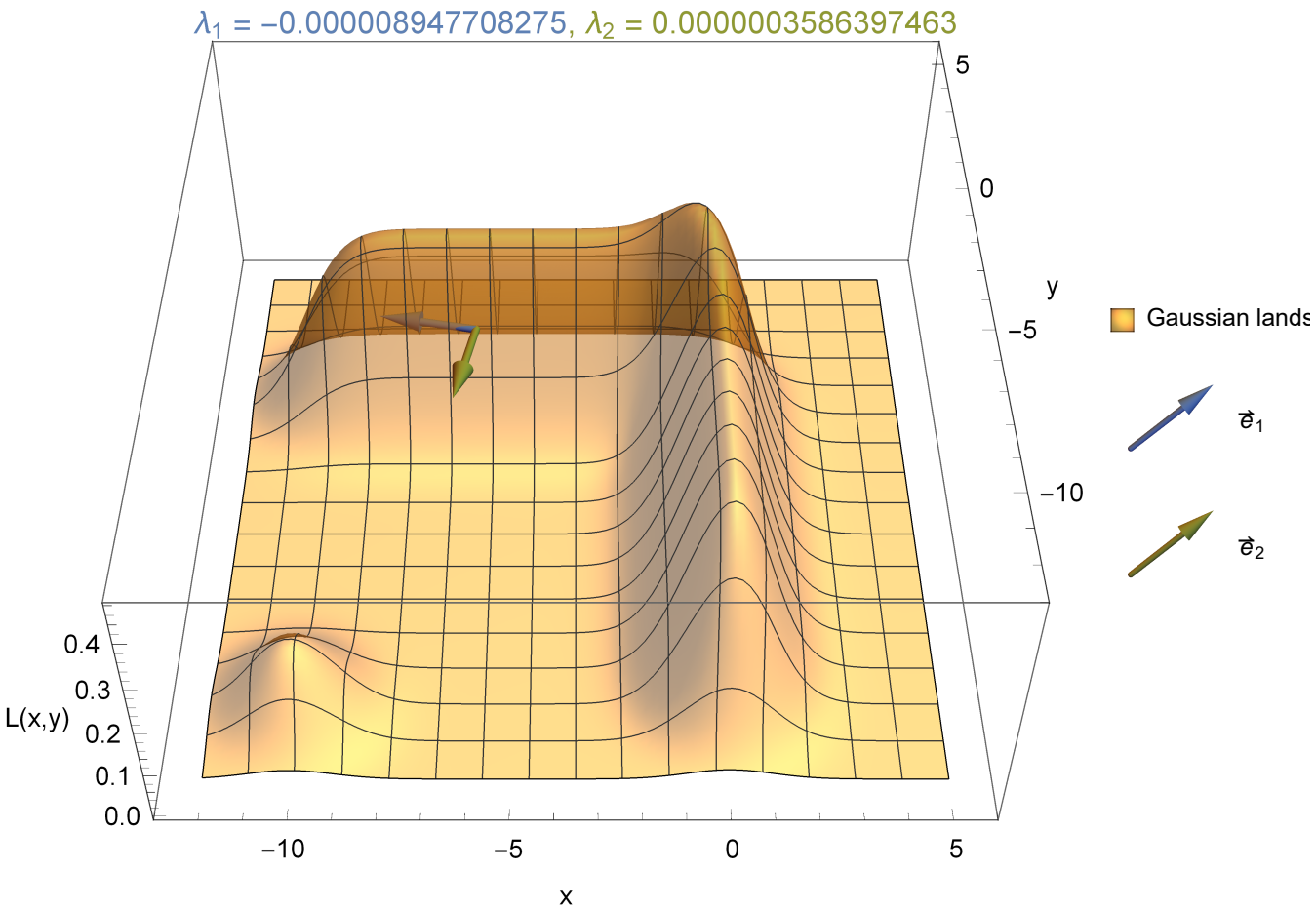


```
plotESlider[x_, y_] := Column[{
    Labeled[
     Slider2D[{x, y}, {{xLeft, yLeft}, {xRight, yRight}}, ImageSize → {150, 150}],
     {"x=" <> ToString[NumberForm[x, 2, SignPadding → True, NumberPadding → {"0", ""}]],
      "y=" <> ToString[NumberForm[y, 2, SignPadding → True, NumberPadding → {"0", ""}]]},
     {Bottom, Left},
     LabelStyle → Directive[FontSize → 14]
    ],
    plotE[{x, y}]
   }, Center, 1, Spacings → 1];

plotESlider[0, -10]
plotESlider[-6, -1]
```










```
(* Note: this takes several hours to complete with 100 plot points *)
(*xStep=1;
yStep=1;
Do[
 Export[
   FileNameJoin[{
     NotebookDirectory[],
     "frames/x="<>IntegerString[Round[x-xLeft/xStep,1],10,2]<>
      "y="<>IntegerString[Round[y-yLeft/yStep,1],10,2]<>".png"}
   ],
   plotESlider[x,y],
   "VideoFrames",
   Antialiasing→True
  ];

 ,{x,xLeft,xRight,xStep},{y,yLeft,yRight,yStep}
]*)

(* 3D locator: https://
 groups.google.com/forum/#!topic/comp.soft-sys.math.mathematica/P0BjXP4zVtY *)
```