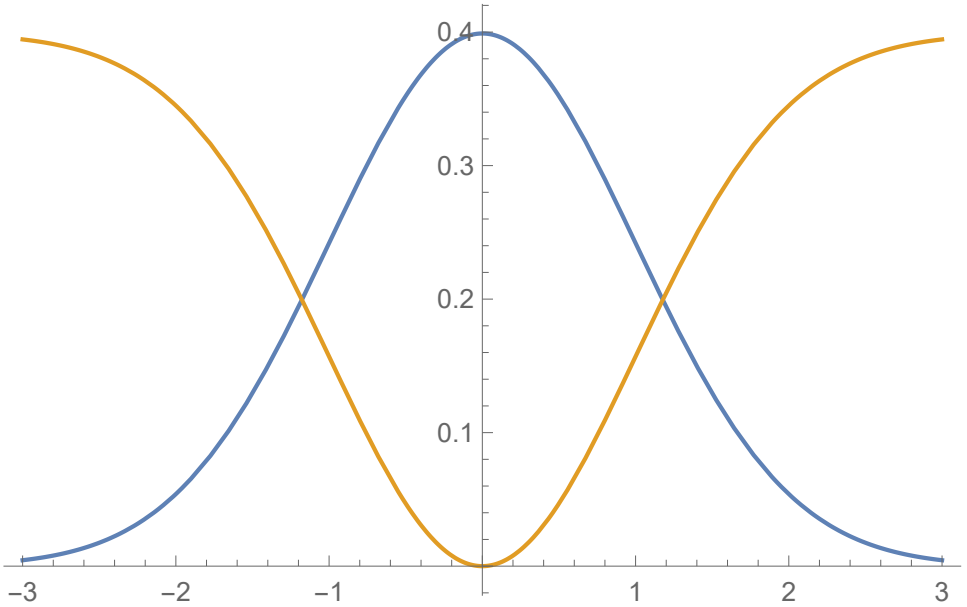
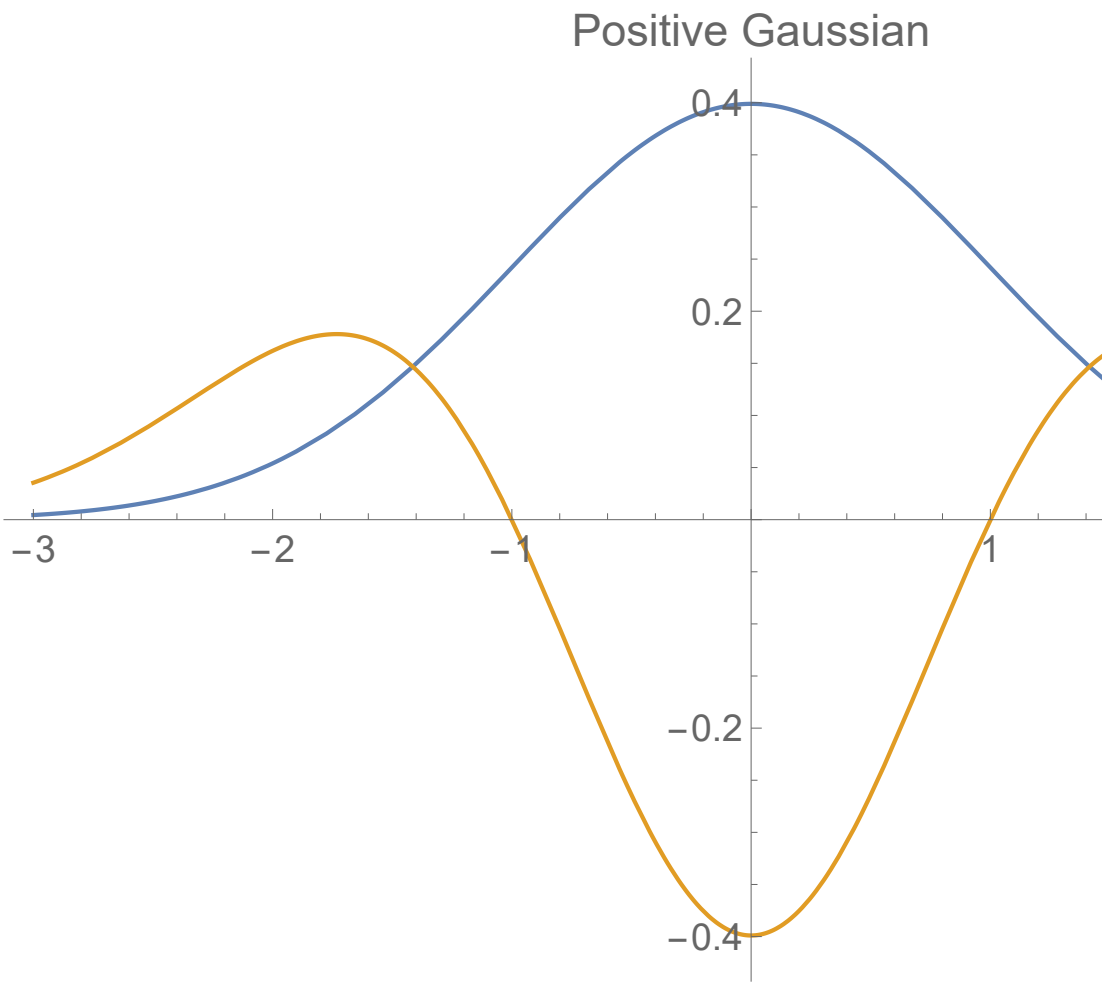
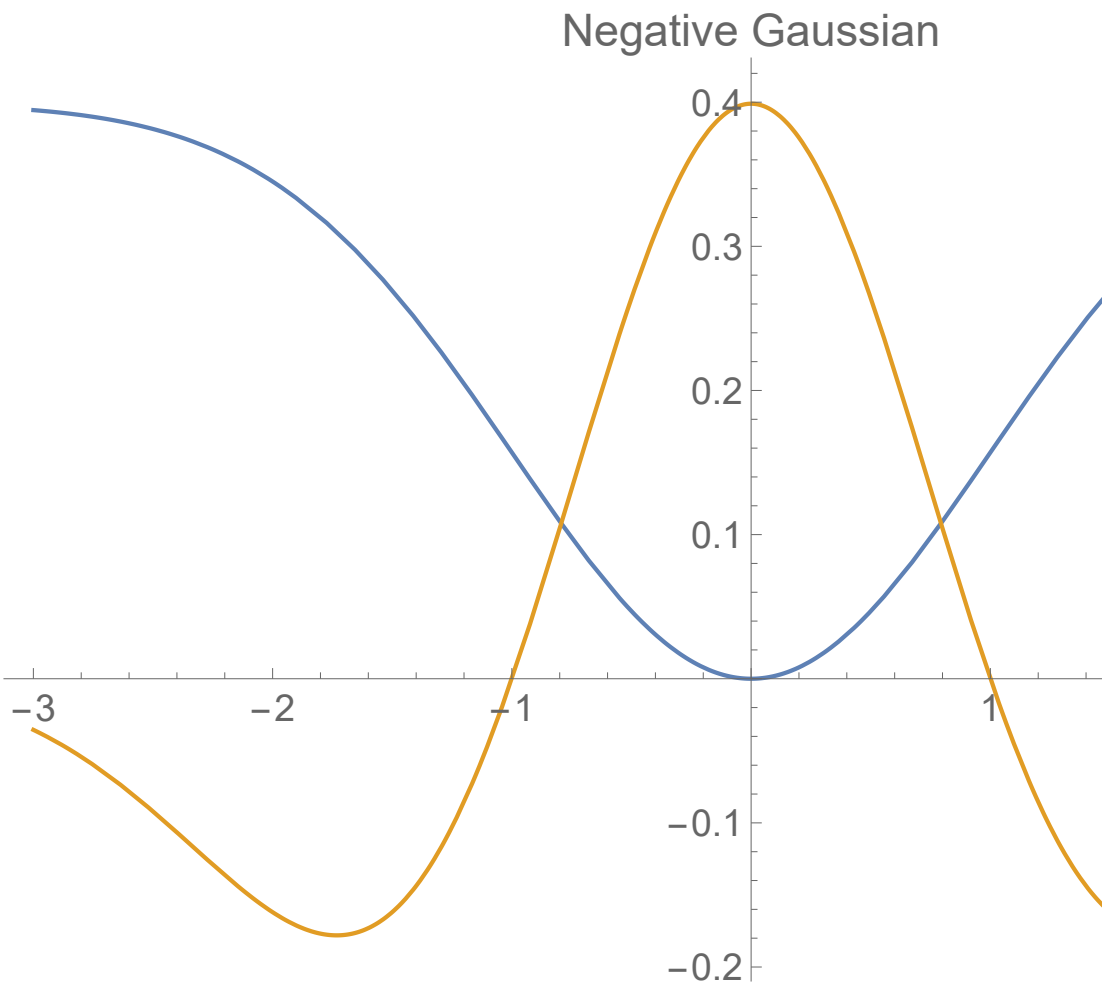
```
gPos[x_] := PDF[NormalDistribution[0, 1], x];
gNeg[x_] := gPos[0] - gPos[x];
g2[x_, y_] :=
  PDF[MultinormalDistribution[{0, 0}, (1 0; 0 1)], {x, y}];
g1[x_, y_] := g2[0, 0] - g2[x, y];
g3[x_, y_] := (gPos[x] + 1) * (gNeg[y] + 1);
Hessian2D[f_] :=
  (D[D[f[x, y], {x}], {x}]  D[D[f[x, y], {x}], {y}]
   D[D[f[x, y], {y}], {x}]  D[D[f[x, y], {y}], {y}]);

xLeft = -3;
xRight = 3;
yLeft = -3;
yRight = 3;

Plot[{gPos[x], gNeg[x]}, {x, xLeft, xRight}]
```




```
pPosGauss = Plot[{gPos[x], gPos''[x]}, {x, xLeft, xRight},
  PlotLegends → {"g(x)", "∂²/∂²x² g(x)"},
  LabelStyle → Directive[FontSize → 14],
  ImageSize → Large, AxesLabel → {"x"},
  PlotLabel → "Positive Gaussian"]
```

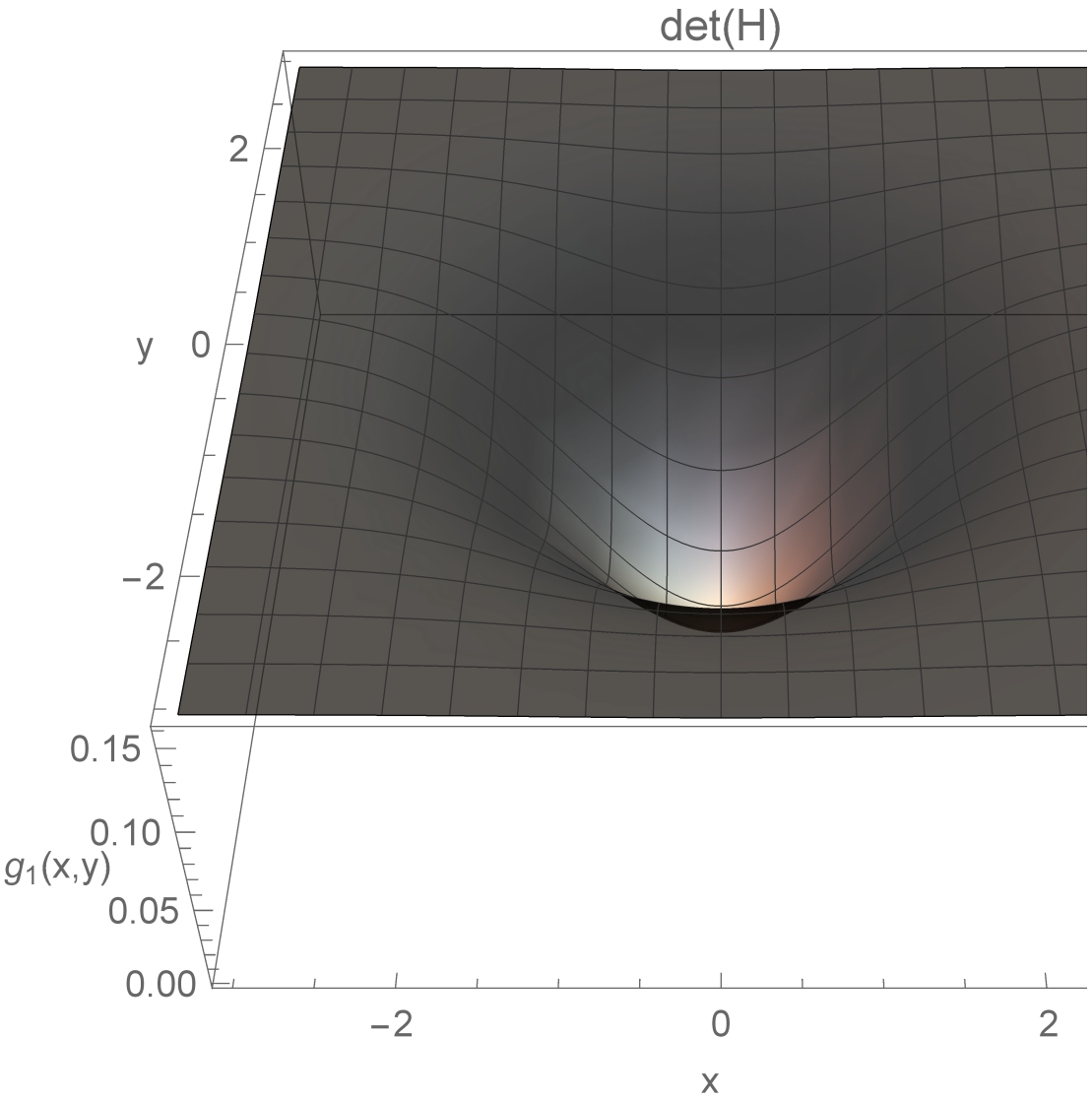



```
pNegGauss = Plot[{gNeg[x], gNeg''[x]}, {x, xLeft, xRight},
  PlotLegends → {"g(0)-g(x)", "∂²/∂²x² (g(0)-g(x))"},
  LabelStyle → Directive[FontSize → 14],
  ImageSize → Large, AxesLabel → {"x"},
  PlotLabel → "Negative Gaussian"]
```

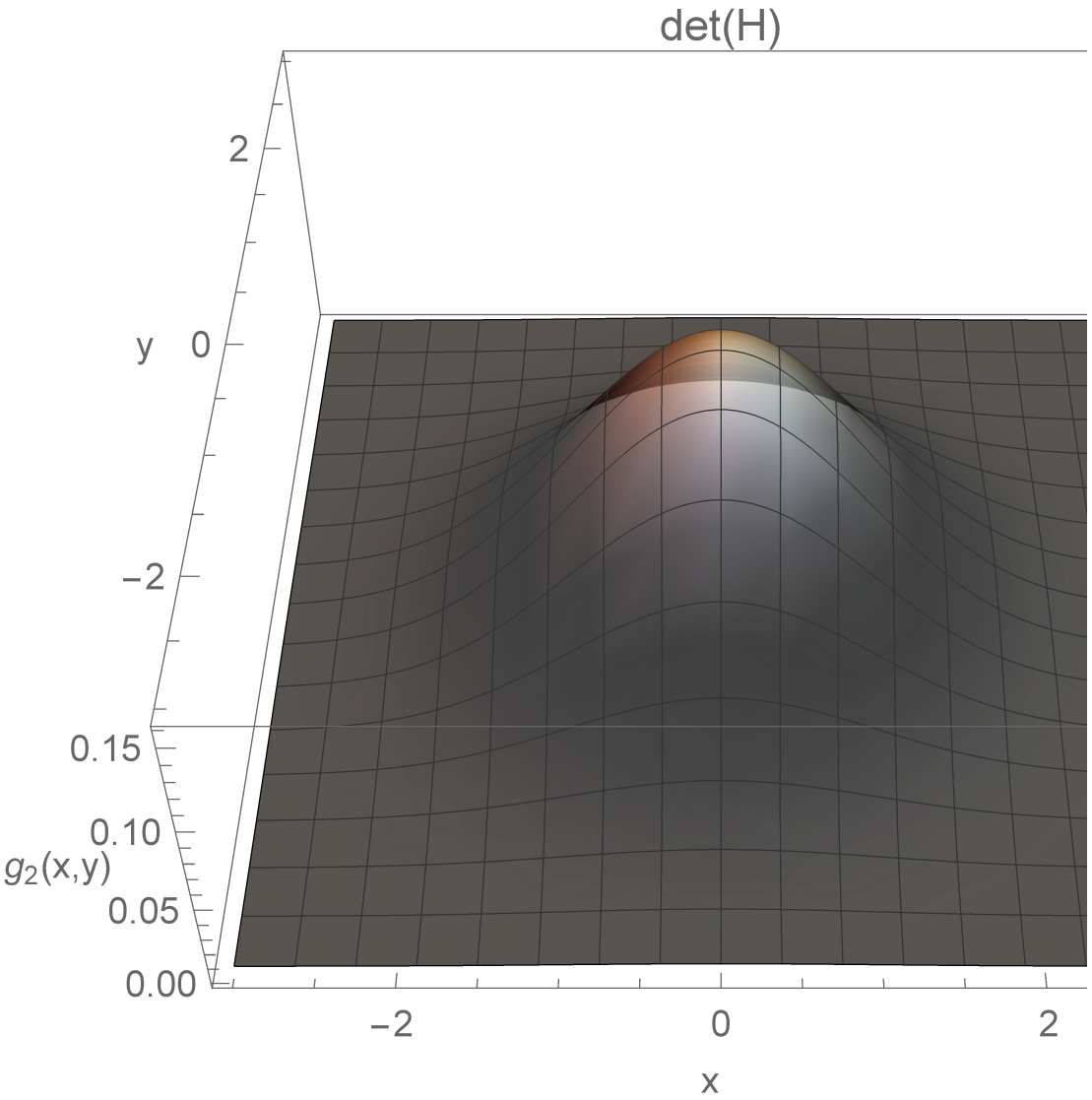

Negative Gaussian
0.4
0.3
0.2
0.1
−0.1
−0.2
−3
−2
−1
1

```
padPlots[plots_] := Block[{widths, paddedPlots},
  paddedPlots = ConstantArray[0, Length[plots]];
  widths =
   Table[ImageDimensions[Rasterize[plots[[i]]]][[1]],
    {i, 1, Length[plots]}];

  Do[
   paddedPlots[[i]] = ImagePad[
      Rasterize[plots[[i]]],
      {{0, Max[widths] - widths[[i]]}, {0, 0}},
      White
     ];
   , {i, 1, Length[plots]}];

  paddedPlots
 ]

(*paddedPlots=padPlots[{pPosGauss,pNegGauss}];
Export[FileNameJoin[{NotebookDirectory[],"type=0.png"}],
 paddedPlots[[1]],"VideoFrames",Antialiasing→True];
Export[FileNameJoin[{NotebookDirectory[],"type=1.png"}],
 paddedPlots[[2]],"VideoFrames",Antialiasing→True];*)

plotDetResp[f_, label_] := Show[
   Legended[
    Plot3D[f[x, y], {x, xLeft, xRight},
     {y, yLeft, yRight},
     AspectRatio → Automatic,
     ImageSize → 490,
     AxesLabel → {"x", "y", label},
     PlotLegends → {"Gaussian function"},
     PlotLabel → "det(H)",
     PlotPoints → 100,
     PerformanceGoal → "Quality",
     BaseStyle → {FontSize → 14},
     ViewPoint → {0, -2, 2},
     PlotStyle → Directive[Texture[
```

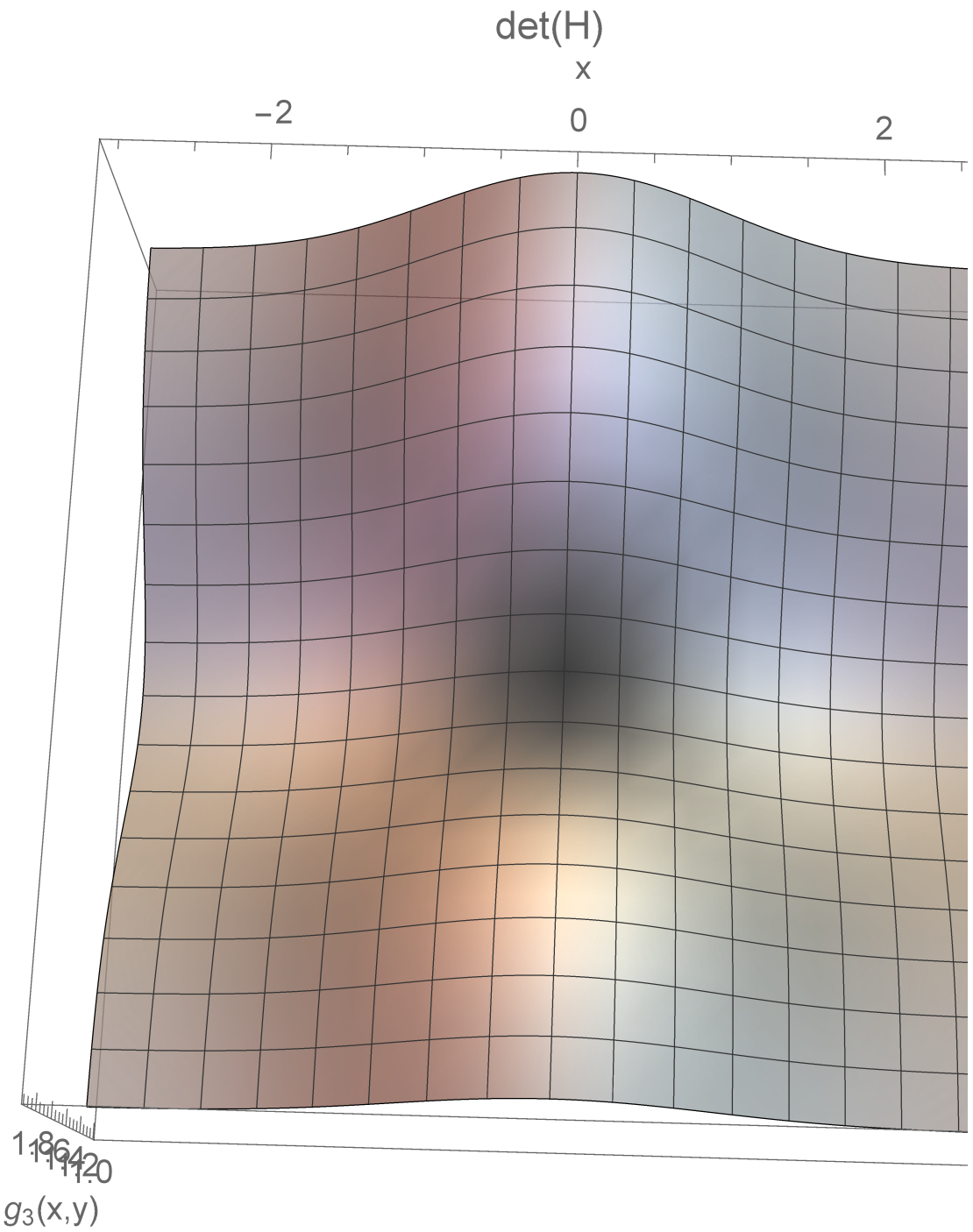
```
            DensityPlot[Det[Hessian2D[f]] // Evaluate,
             {x, xLeft, xRight}, {y, yLeft, yRight},
             ColorFunction → GrayLevel,
             PlotRange → Full,
             Frame → None,
             PlotRangePadding → 0
            ]
           ], Opacity[0.75]]
        ],
        BarLegend[{
          GrayLevel,
          (* Creates a density plot with the color
           bar and extracts the determinant response
           range. The plot itself is not used further *)
          InputForm[DensityPlot[
             Det[Hessian2D[f]] // Evaluate,
             {x, xLeft, xRight}, {y, yLeft, yRight},
             PlotLegends → Automatic,
             ColorFunction → GrayLevel, PlotRange → Full]][[
           1, 2, 1, 1, 2]]
         },
         LegendLayout → "Column"
        ]
       ]
      ];

plotDetResp[g1, "g1(x,y)"]
```




```
plotDetResp[g2, "g₂(x,y)"]
```




```
plotDetResp[g3, "g3(x,y)"]
```


det(H)
x
−2
0
2
g_3(x,y)