HW#13 최적화 실습

In[6]:=
```
f [x_, y_] = Sin[x + y - 1] + (x - y - 1) ^2 - 1.5 * x + 2.5 * y + 1
f = Function[{x, y}, Sin[x + y - 1] + (x - y - 1) ^2 - 1.5 * x + 2.5 * y + 1][x, y]
```

Out[6]= $1 - 1.5\,x + (-1 + x - y)^2 + 2.5\,y - \text{Sin}[1 - x - y]$

Out[7]= $1 - 1.5\,x + (-1 + x - y)^2 + 2.5\,y - \text{Sin}[1 - x - y]$

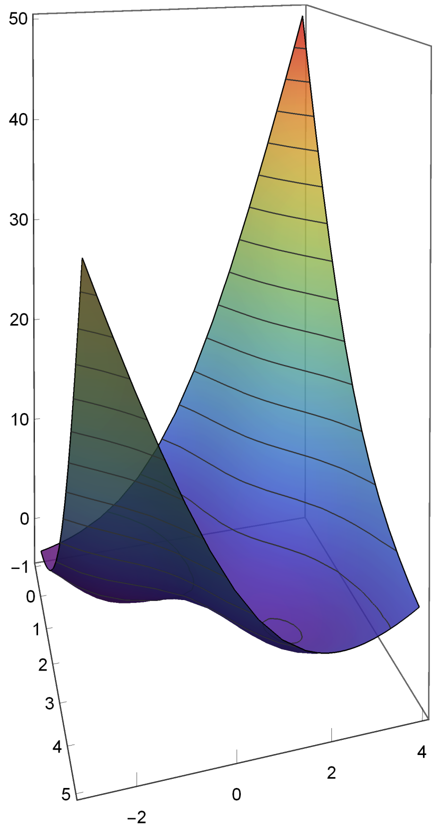
In[8]:=
```
grad = Grad[f, {x, y}]
df[x_, y_] = grad
```

Out[8]= $\{-1.5 + 2\,(-1 + x - y) + \text{Cos}[1 - x - y],\ 2.5 - 2\,(-1 + x - y) + \text{Cos}[1 - x - y]\}$

Out[9]= $\{-1.5 + 2\,(-1 + x - y) + \text{Cos}[1 - x - y],\ 2.5 - 2\,(-1 + x - y) + \text{Cos}[1 - x - y]\}$

In[10]:=
```
Plot3D[f, {x, -1, 5}, {y, -3, 4}, PlotRange → All, ClippingStyle → None, AspectRatio → 2,
 PlotTheme → "Web", PlotStyle → Opacity[.85], ColorFunction → "Rainbow"]
```

Out[10]=



In[538]:=
```
p = {RandomReal[{-1, 5}], RandomReal[{-3, 4}]}
```

Out[538]= {-0.955958, 1.46699}

Gradient Descent

In[539]:=
```
lamda = 0.1
x = p[[1]]
y = p[[2]]
pts = {};
For[i = 0, i < 20, i++, {x, y} = {x, y} - lamda * df[x, y]; pts = Append[pts, {x, y}];]
pts = Join[{{p[[1]], p[[2]]}}, pts];
pts // MatrixForm
ContourPlot[Sin[x + y - 1] + (x - y - 1)^2 - 1.5 * x + 2.5 * y + 1, {x, -3, 5}, {y, -3, 4},
 Contours → 10, ContourLabels → True, Epilog → {Red, Line[pts], Point[pts]}]
```

Out[539]= 0.1

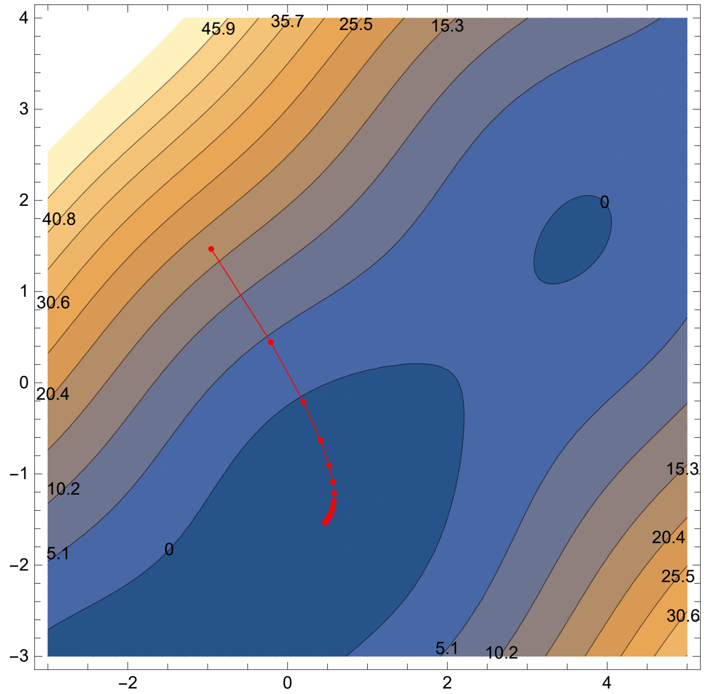
Out[540]= -0.955958

Out[541]= 1.46699

Out[545]//MatrixForm=

$$\begin{pmatrix} -0.955958 & 1.46699 \\ -0.209651 & 0.444118 \\ 0.199002 & -0.208737 \\ 0.414246 & -0.630398 \\ 0.520591 & -0.906195 \\ 0.56682 & -1.08925 \\ 0.580771 & -1.21287 \\ 0.578169 & -1.29802 \\ 0.567782 & -1.35793 \\ 0.554399 & -1.40103 \\ 0.540549 & -1.43271 \\ 0.527484 & -1.45647 \\ 0.515751 & -1.47462 \\ 0.505517 & -1.48871 \\ 0.496753 & -1.49978 \\ 0.489336 & -1.50858 \\ 0.483109 & -1.51564 \\ 0.477909 & -1.52134 \\ 0.473582 & -1.52597 \\ 0.469991 & -1.52974 \\ 0.467015 & -1.53282 \end{pmatrix}$$

Out[546]=



In[365]:=
```
lamda = 0.01
x = p[[1]];
y = p[[2]];
pts = {};
For[i = 0, i < 20, i++, {x, y} = {x, y} - lamda * df[x, y]; pts = Append[pts, {x, y}];]
pts = Join[{{p[[1]], p[[2]]}}, pts];
pts // MatrixForm;
ContourPlot[Sin[x + y - 1] + (x - y - 1)^2 - 1.5 * x + 2.5 * y + 1, {x, -3, 5}, {y, -3, 4},
  Contours → 10, ContourLabels → True, Epilog → {Red, Line[pts], Point[pts]}];
```

Out[365]= 0.01

Newton's 방법

In[242]:=
```
Clear[x]
Clear[y]
d2f[x_, y_] = Grad[df[x, y], {x, y}]
```

Out[244]= {{2 + Sin[1 - x - y], -2 + Sin[1 - x - y]}, {-2 + Sin[1 - x - y], 2 + Sin[1 - x - y]}}

In[547]:=
```
x = p[[1]]
y = p[[2]]
pts = {};
For[i = 0, i < 5, i++,
 {x, y} = {x, y} - Dot[df[x, y], Inverse[d2f[x, y]]]; pts = Append[pts, {x, y}];]
pts = Join[{{p[[1]], p[[2]]}}, pts];
pts // MatrixForm
ContourPlot[Sin[x + y - 1] + (x - y - 1)^2 - 1.5 * x + 2.5 * y + 1, {x, -3, 5}, {y, -3, 4},
 Contours → 10, ContourLabels → True, Epilog → {Red, Line[pts], Point[pts]}]
```

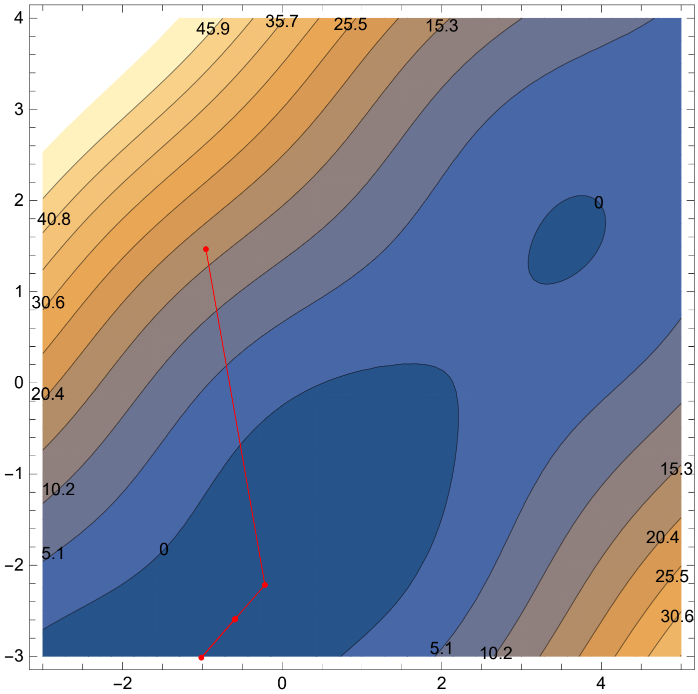
Out[547]= -0.955958

Out[548]= 1.46699

Out[552]//MatrixForm=

$$\begin{pmatrix} -0.955958 & 1.46699 \\ -0.216456 & -2.21646 \\ -1.01353 & -3.01353 \\ -0.587878 & -2.58788 \\ -0.59442 & -2.59442 \\ -0.594395 & -2.5944 \end{pmatrix}$$

Out[553]=



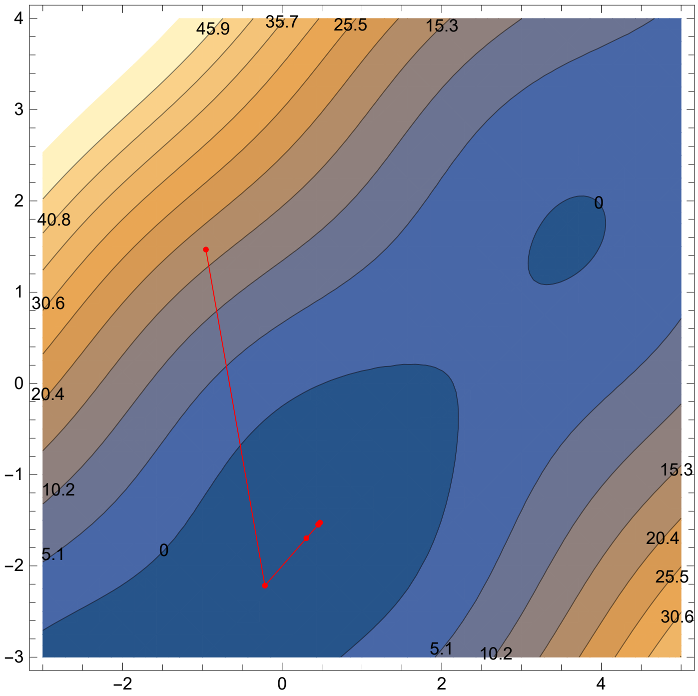
Newton + TrustRegion

In[554]:=
```
j = 1;
pts = Reap[FindMinimum[Sin[x + y - 1] + (x - y - 1)^2 - 1.5 * x + 2.5 * y + 1,
    {{x, p[[1]]}, {y, p[[2]]}}, Method → {"Newton", "StepControl" → "TrustRegion"},
    StepMonitor :→ Print["Step:", j++, " x, y =", Sow[{x, y}]]]][[2, 1]]
pts = Join[{{p[[1]], p[[2]]}}, pts];
ContourPlot[Sin[x + y - 1] + (x - y - 1)^2 - 1.5 * x + 2.5 * y + 1, {x, -3, 5}, {y, -3, 4},
 Contours → 10, ContourLabels → True, Epilog → {Red, Line[pts], Point[pts]}]
```

```
Step:1 x, y ={-0.216456, -2.21646}
Step:2 x, y ={0.303489, -1.69651}
Step:3 x, y ={0.474415, -1.52558}
Step:4 x, y ={0.453053, -1.54695}
Step:5 x, y ={0.452802, -1.5472}
Step:6 x, y ={0.452802, -1.5472}
```

Out[555]=
```
{{-0.216456, -2.21646}, {0.303489, -1.69651}, {0.474415, -1.52558},
 {0.453053, -1.54695}, {0.452802, -1.5472}, {0.452802, -1.5472}}
```

Out[557]=



Newton + LineSearch

In[558]:=
```
j = 1;
pts = Reap[FindMinimum[Sin[x + y - 1] + (x - y - 1)^2 - 1.5 * x + 2.5 * y + 1,
     {{x, p[[1]]}, {y, p[[2]]}}, Method → {"Newton", "StepControl" → "LineSearch"},
     StepMonitor :→ Print["Step:", j++, " x, y =", Sow[{x, y}]]]][[2, 1]]
pts = Join[{{p[[1]], p[[2]]}}, pts];
ContourPlot[Sin[x + y - 1] + (x - y - 1)^2 - 1.5 * x + 2.5 * y + 1, {x, -3, 5}, {y, -3, 4},
 Contours → 10, ContourLabels → True, Epilog → {Red, Line[pts], Point[pts]}]
```

```
Step:1 x, y ={-0.216456, -2.21646}

Step:2 x, y ={0.192783, -1.66997}

Step:3 x, y ={0.494366, -1.50563}

Step:4 x, y ={0.453664, -1.54634}

Step:5 x, y ={0.452803, -1.5472}

Step:6 x, y ={0.452802, -1.5472}
```

Out[559]= {{-0.216456, -2.21646}, {0.192783, -1.66997}, {0.494366, -1.50563}, {0.453664, -1.54634}, {0.452803, -1.5472}, {0.452802, -1.5472}}

Out[561]=