# Minimize

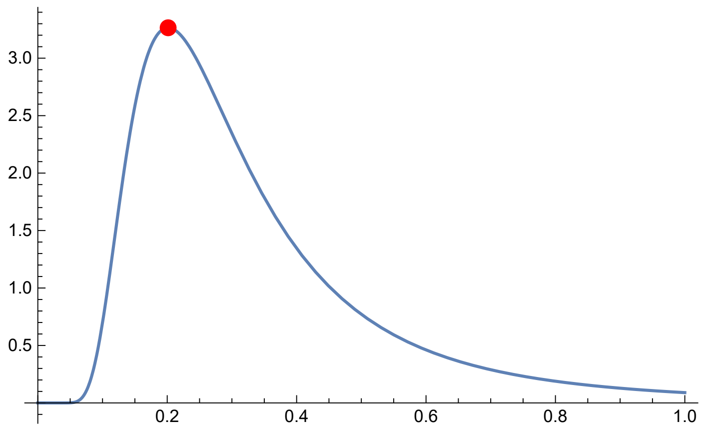
In[∘]:=
$$f = \frac{15}{\pi^4 \left(x^5 \left(\mathrm{Exp}\left[\frac{1}{x}\right] - 1\right)\right)};$$

In[∘]:=
```
{max, val} = Maximize[{f, x ≥ 0}, x]
```

Out[∘]=
$$\left\{\frac{15\left(5+\mathrm{ProductLog}\left[-\frac{5}{e^5}\right]\right)^5}{\left(-1+e^{5+\mathrm{ProductLog}\left[-\frac{5}{e^5}\right]}\right)\pi^4}, \left\{x \to \frac{1}{5+\mathrm{ProductLog}\left[-\frac{5}{e^5}\right]}\right\}\right\}$$

In[∘]:=
```
Plot[f, {x, 0, 1}, Epilog → {Red, PointSize[0.025], Point[{x /. val, max}]}]
```

General: $2.81067 \times 10^{23}$ $1.01309096570 \times 10^{-21258}$ is too small to represent as a normalized machine number; precision may be lost.

Out[∘]=



# Linear Programming

*c* is a linear objective f(x)
*m* is a matrix of linear constraints
*b* are bounds for the constraints

In[∘]:=
```
c = {-1, -2};
```

$$m = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ -1 & -1 \\ -1 & 1 \\ 0 & 1 \\ 0 & -1 \end{pmatrix};$$

In[∘]:=
```
b = {0, 0, -1, -1, -0.25, -0.25};
```

In[∘]:=
```
sol = LinearProgramming[c, m, b]
```

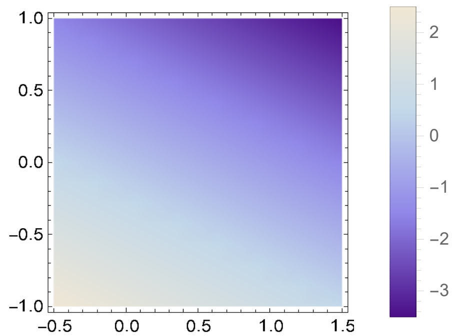
Out[∘]=
```
{0.75, 0.25}
```

```
In[•]:= Style[constr = And @@ MapThread[GreaterEqual, {m.{x, y}ᵀ, b}], Blue, Bold]
```

Out[•]= $x + y \geq 0 \,\&\&\, x - y \geq 0 \,\&\&\, -x - y \geq -1 \,\&\&\, -x + y \geq -1 \,\&\&\, y \geq -0.25 \,\&\&\, -y \geq -0.25$

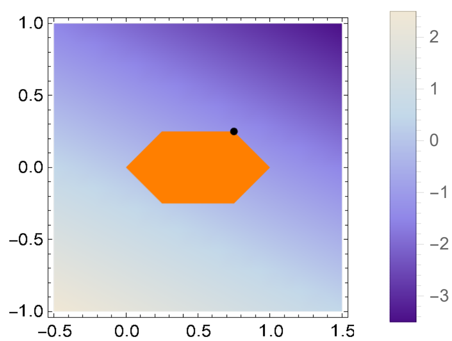
```
In[•]:= objplot = DensityPlot[c.{x, y}, {x, -0.5, 1.5}, {y, -1, 1},
          ColorFunction → "LakeColors", ImageSize → Small, PlotLegends → True]
```

Out[•]=




```
In[•]:= Show[objplot,
         RegionPlot[constr, {x, -.5, 1.5}, {y, -1, 1},
          BoundaryStyle → None, PlotStyle → Orange, ImageSize → Small],
         Graphics[{PointSize[0.025], Point@sol}]
        ]
```

Out[•]=




## Exact linear programs

```
In[•]:= c = {1, 1}; m = {{π, 2}}; b = {3};

In[•]:= LinearProgramming[c, m, b]
```

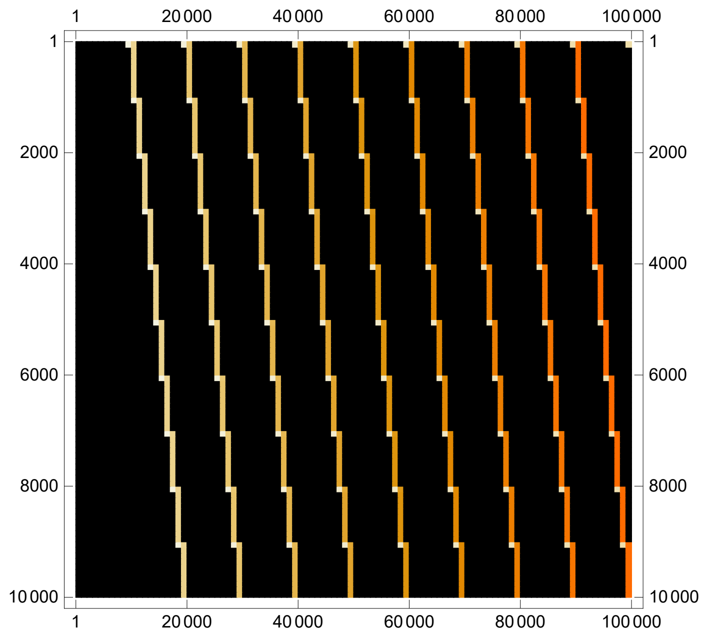
Out[•]= $\left\{\frac{3}{\pi}, 0\right\}$

## Large scale linear programs

```
In[•]:= c = Range[10^5];

In[•]:= m = SparseArray[Table[Band[{1, j 10^4}] → N[j], {j, 1, 10}], {10^4, 10^5}];
```

In[∘]:=
```
MatrixPlot[m, ColorRules → (0 → Black)]
```
Out[∘]=




In[∘]:=
```
b = Range[10^4];
```

In[∘]:=
```
sol = LinearProgramming[c, m, b];
```

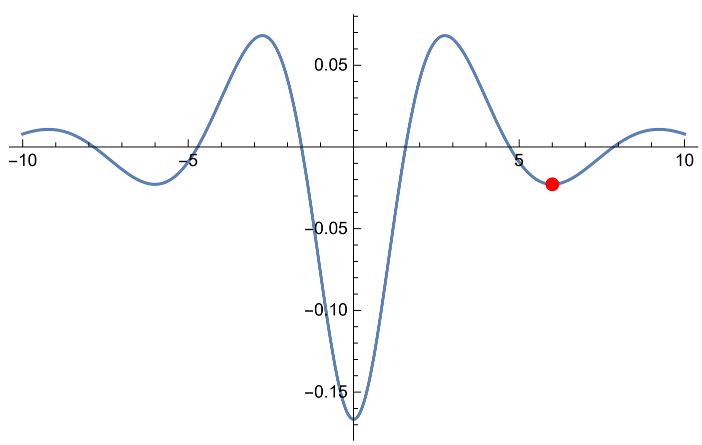
# Local optimization

In[∘]:=
```
f = -Cos[x] / (x^2 + 6);
```

In[∘]:=
```
{min, val} = FindMinimum[f, {x, 5}]
```

Out[∘]=
```
{-0.0228615, {x → 6.00504}}
```

In[∘]:=
```
Plot[f, {x, -10, 10}, PlotRange → All,
 Epilog → {Red, PointSize[0.02], Point[{x /. val, min}]}]
```
Out[∘]=

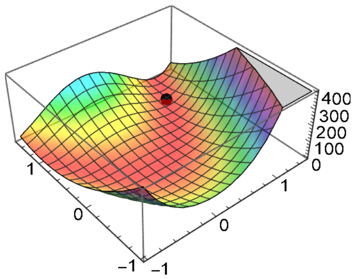


In[∘]:=
```
{min, {x1, x2}} = FindMinimum[(1 - X1)^2 + 100 (-X1^2 + X2)^2, {{X1, -1.2`}, {X2, -1.2`}}]
```

Out[∘]=
```
{0., {X1 → 1., X2 → 1.}}
```

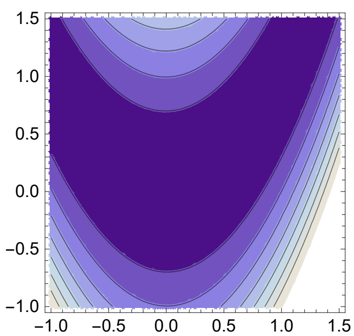
In[∘]:=
```
Show[Plot3D[(1 - X1)^2 + 100 (-X1^2 + X2)^2, {X1, -1, 1.5},
  {X2, -1, 1.5}, PlotStyle → Opacity[0.6], ColorFunction → (Hue[#3] &)],
 Graphics3D[{Black, PointSize[0.04], Point[{X1 /. x1, X2 /. x2, min}]}], ImageSize → Small]
```

Out[∘]=




In[∘]:=
```
ContourPlot[(1 - X1)^2 + 100 (-X1^2 + X2)^2, {X1, -1, 1.5},
 {X2, -1, 1.5}, ColorFunction → "LakeColors", ImageSize → Small]
```

Out[∘]=



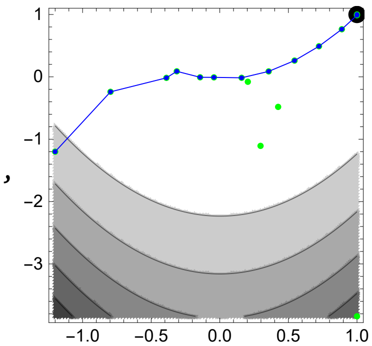

In[∘]:=
```
FindMinimum[(1 - X1)^2 + 100 (-X1^2 + X2)^2,
 {{X1, -1.2`}, {X2, -1.2`}}, StepMonitor :→ Print["(X1,X2): ", {X1, X2}]]
```

```
(X1,X2): {-0.796577, -0.240194}
(X1,X2): {-0.389079, -0.0162315}
(X1,X2): {-0.313527, 0.0887007}
(X1,X2): {-0.14356, -0.0072256}
(X1,X2): {-0.0427052, -0.00832815}
(X1,X2): {0.15899, -0.0153524}
(X1,X2): {0.356243, 0.0874354}
(X1,X2): {0.544862, 0.260647}
(X1,X2): {0.722809, 0.490289}
(X1,X2): {0.889576, 0.763313}
(X1,X2): {1., 0.987807}
(X1,X2): {1., 1.}
```

Out[∘]=
```
{0., {X1 → 1., X2 → 1.}}
```

In[∘]:=
```
<< Optimization`UnconstrainedProblems`
```

In[∘]:=
```
FindMinimumPlot[(1 - X1)^2 + 100 (-X1^2 + X2)^2, {{X1, -1.2`}, {X2, -1.2`}}]
```

Out[∘]=
```
{{0., {X1 → 1., X2 → 1.}}, {Steps → 12, Function → 17},
```



```
}
```

# Global optimization

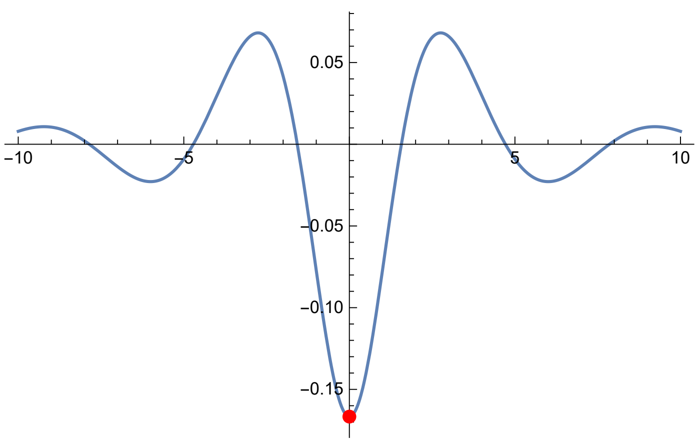
In[∘]:=
```
f = -Cos[x] / (x^2 + 6);
```

In[∘]:=
```
{min, val} = NMinimize[f, {x}, Method → "NelderMead"]
```

Out[∘]= $\{-0.166667, \{x \to -3.30872 \times 10^{-24}\}\}$

In[∘]:=
```
Plot[f, {x, -10, 10}, PlotRange → All,
  Epilog → {Red, PointSize[0.02], Point[{x /. val, min}]}]
```

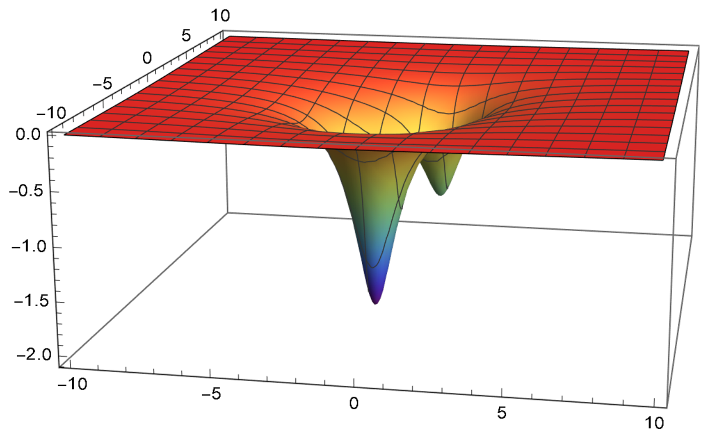
Out[∘]=



# Constrained optimization

In[•]:=
```
Plot3D[-1/((x - 1)^2 + (y - 1)^2 + 1) - 2/((x + 1)^2 + (y + 2)^2 + 1), {x, -10, 10}, {y, -10, 10},
  PlotPoints → 50, PlotRange → All, ColorFunction → "Rainbow", ImageSize → Medium]
```

Out[•]=




In[•]:=
```
f = {-1/((x - 1)^2 + (y - 1)^2 + 1) - 2/((x + 1)^2 + (y + 2)^2 + 1), Abs[x] + Abs[y] > 1};
```

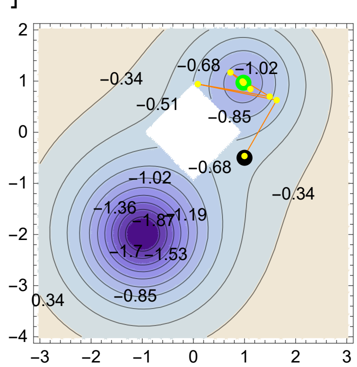
## With FindMinimum

In[•]:=
```
pts = Reap@FindMinimum[f, {{x, 1}, {y, -0.5}}, StepMonitor :→ Sow[{x, y}]]
```

Out[•]=
```
{{-1.14425, {x → 0.978937, y → 0.968405}},
 {{{1., -0.5}, {0.999387, -0.464911}, {1.62823, 0.626921},
   {0.0858557, 0.941013}, {1.49306, 0.696747}, {0.724711, 1.16848},
   {1.11534, 0.843606}, {0.961082, 0.992229}, {0.980545, 0.969958},
   {0.979024, 0.968494}, {0.978938, 0.968406}, {0.978937, 0.968405}}}}
```

In[•]:=
```
ContourPlot[f[[1]], {x, -3, 3}, {y, -4, 2}, ColorFunction → "LakeColors",
 RegionFunction → (Abs[#1] + Abs[#2] ≥ 1 &), Contours → 10, ContourLabels → True,
 Epilog → {Orange, Line@pts[[2, 1]], Black, PointSize[0.05], Point@pts[[2, 1, 1]],
   Green, Point@pts[[2, 1, -1]], PointSize[0.02], Yellow, Point /@ Rest[pts[[2, 1]]]},
 ImageSize → Small
]
```

Out[•]=



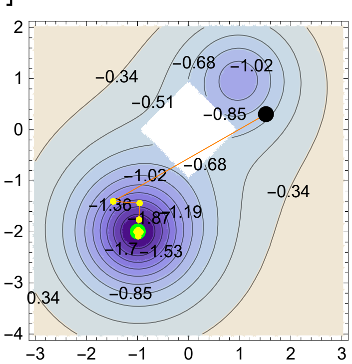
### With NMinimize

In[ ]:=
```
pts = Reap@NMinimize[f, {{x, -2, 2}, {y, -2, 2}}, StepMonitor :> Sow[{x, y}]]
```

Out[ ]=
```
{{-2.0716, {x → -0.994861, y → -1.99229}},
 {{{1.51442, 0.304803}, {-1.47574, -1.40263}, {-1.47574, -1.40263},
   {-1.47574, -1.40263}, {-1.47574, -1.40263}, {-1.47574, -1.40263},
   {-1.47574, -1.40263}, {-0.95838, -1.43686}, {-0.95838, -1.43686},
   {-0.971999, -1.76026}, {-0.971999, -1.76026}, {-0.971999, -1.76026},
   {-0.985617, -2.08366}, {-0.985617, -2.08366}, {-0.985617, -2.08366},
   {-0.941688, -2.00295}, {-0.986759, -1.9664}, {-0.986759, -1.9664},
   {-1.01999, -1.99762}, {-0.989148, -2.00809}, {-0.995665, -1.98463},
   {-0.995665, -1.98463}, {-0.99504, -1.99945}, {-0.989929, -1.98956},
   {-0.994075, -1.98957}, {-0.993521, -1.99451}, {-0.995732, -1.99327},
   {-0.994351, -1.99173}, {-0.994351, -1.99173}, {-0.995404, -1.99244},
   {-0.994415, -1.99238}, {-0.99463, -1.99207}, {-0.994963, -1.99233},
   {-0.994963, -1.99233}, {-0.994963, -1.99233}, {-0.994963, -1.99233},
   {-0.994837, -1.99225}, {-0.994839, -1.99233}, {-0.994839, -1.99233},
   {-0.994854, -1.99228}, {-0.994854, -1.99228}, {-0.994864, -1.99229},
   {-0.994861, -1.99229}, {-0.994861, -1.99229}}}}
```

In[ ]:=
```
ContourPlot[f[[1]], {x, -3, 3}, {y, -4, 2}, ColorFunction → "LakeColors",
 RegionFunction → (Abs[#1] + Abs[#2] ≥ 1 &), Contours → 10, ContourLabels → True,
 Epilog → {Orange, Line@pts[[2, 1]], Black, PointSize[0.05], Point@pts[[2, 1, 1]],
   Green, Point@pts[[2, 1, -1]], PointSize[0.02], Yellow, Point /@ Rest[pts[[2, 1]]]},
 ImageSize → Small
]
```

Out[ ]=



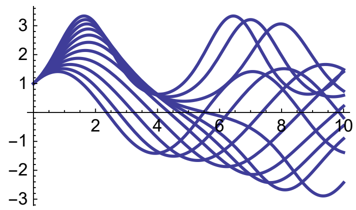

# Constrained optimization

find the value of a for which the solution of y’’(t)+y(t)=3 a sin(y(t)), y(0)=y’(0)=1 has a minimal arc length from t=0 to t=10

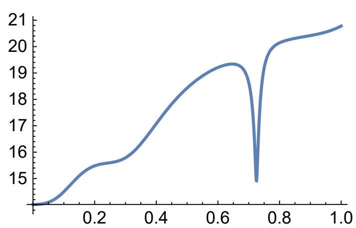
In[∘]:=
```
psol = ParametricNDSolve[{y''[t] + y[t] == 3 a Sin[y[t]], y[0] == y'[0] == 1,
   α'[t] == Sqrt[1 + y[t]^2], α[0] == 0}, {y, α}, {t, 0, 10}, {a}]
```

Out[∘]=
```
{y → ParametricFunction[ Expression: y  Parameters: {a} ],
 α → ParametricFunction[ Expression: α  Parameters: {a} ]}
```

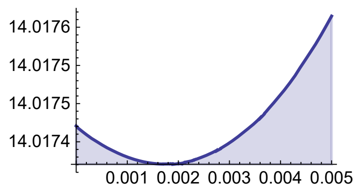
In[∘]:=
```
Plot[Table[Evaluate[y[a][t] /. psol], {a, 0, 1, 0.1}],
 {t, 0, 10}, PlotStyle → (ColorData[1][1]), ImageSize → Small]
```

Out[∘]=



In[∘]:=
```
Plot[Evaluate[α[a][10] /. psol], {a, 0, 1}, ImageSize → Small]
```

Out[∘]=



In[∘]:=
```
Plot[Evaluate[α[a][10] /. psol], {a, 0, 0.005},
 ImageSize → Small, PlotStyle → ColorData[1][1], Filling → Bottom]
```

Out[∘]=


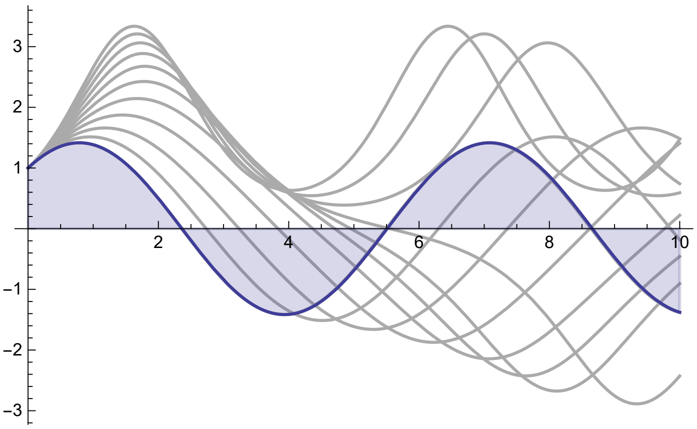

## global minima

In[∘]:=
```
gmin0 = NMinimize[{α[a][10] /. psol, 0. ≤ a ≤ 0.005}, {a}, AccuracyGoal → 7]
```

Out[∘]=
```
{14.0174, {a → 0.00189029}}
```

In[◦]:=
```
Show[
 Plot[Table[Evaluate[y[a][t] /. psol], {a, 0, 1, 0.1}],
  {t, 0, 10}, PlotStyle → Lighter@Gray],
 Plot[Evaluate[y[a /. gmin0[[-1]] ][t] /. psol], {t, 0, 10},
  PlotStyle → ColorData[1][1], Filling → Axis]
]
```

Out[◦]=



## local minima

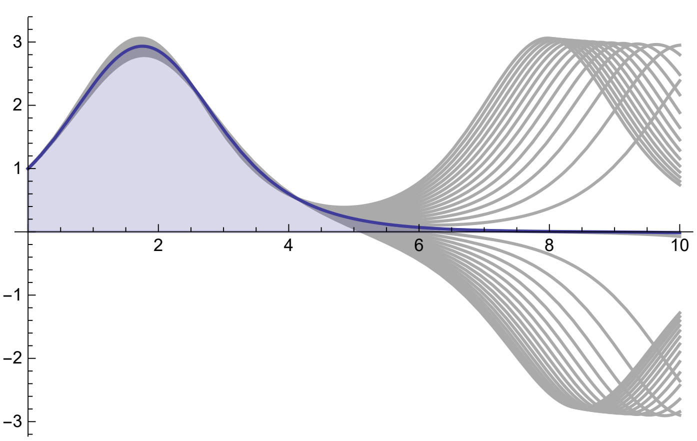
In[◦]:=
```
lmin0 = FindMinimum[{α[a][10] /. psol, 0 < a < 1}, {a, 0.7}, AccuracyGoal → 7]
```

Out[◦]= {14.9146, {a → 0.725089}}

In[◦]:=
```
fns = Table[Evaluate[y[a][t] /. psol], {a, 0.65, 0.8, 0.005}];
```

In[◦]:=
```
Show[
 Plot[fns, {t, 0, 10}, PlotStyle → Lighter@Gray],
 Plot[Evaluate[y[a /. lmin0[[-1]] ][t] /. psol],
  {t, 0, 10}, PlotStyle → ColorData[1][1], Filling → Axis]
]
```

Out[◦]=

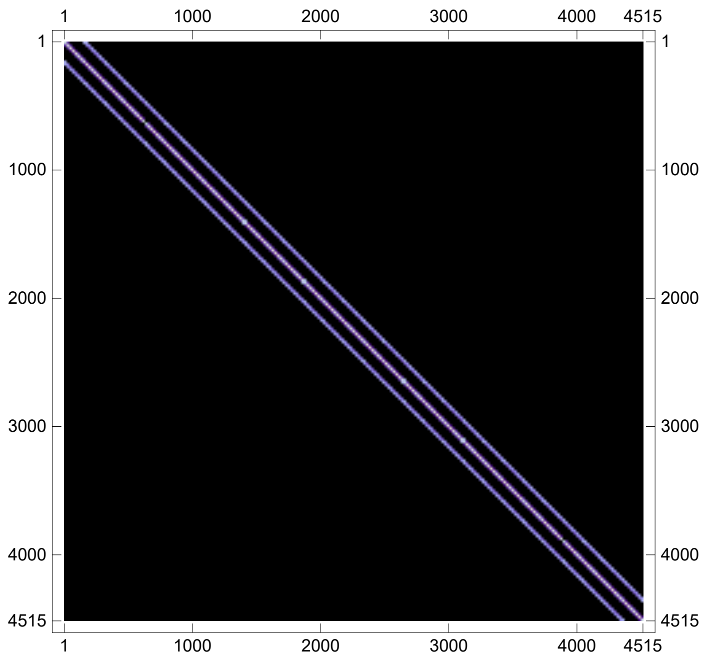


# Linear Solve

In[∘]:= A = ExampleData[{"Matrix", "Boeing/msc04515"}];

In[∘]:= MatrixPlot[A, ColorFunction → "LakeColors", ColorRules → (0 → Black)]

Out[∘]=



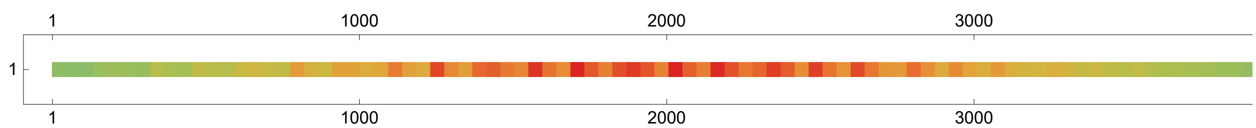

In[∘]:= b = RandomReal[1, Length[A]];

In[∘]:= AbsoluteTiming[sol = LinearSolve[A, b];]

Out[∘]= {0.0767641, Null}

In[∘]:= MatrixPlot[{sol}, ColorFunction → "Rainbow"]

Out[∘]=




In[∘]:= Sn = Normal[A];

In[∘]:= N[ByteCount[Sn] / ByteCount[A]]

Out[∘]= 101.858

In[∘]:= AbsoluteTiming[ex = LinearSolve[Sn, b];]

Out[∘]= {0.652116, Null}

In[ ]:= 
```
ListPlot[ex - sol, PlotRange → All]
```

Out[ ]=