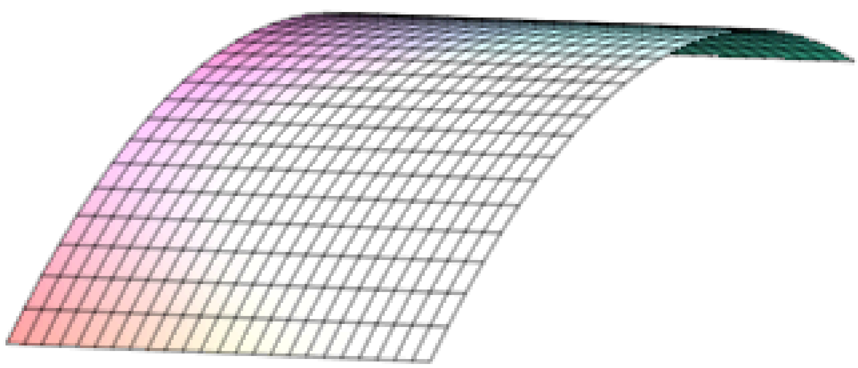
Example with curved constraints

Error, missing operator or `;`

```
> restart:
> with(plots):
  with(plottools):
  with(LinearAlgebra):
> f := (x,y) -> 3 + y-x^2/2;
```

$$f := (x, y) \rightarrow 3 + y - \frac{1}{2} x^2 \quad (1)$$

```
> plot3d( f(x,y), x=-2..2, y=-2..2) ;
```



Curved contraint

```
> hc := (x,y) -> y - x^2 ;
```

$$hc := (x, y) \rightarrow y - x^2 \quad (2)$$

Straight contraint

```
> hs := (x,y) -> y ;
```

$$hs := (x, y) \rightarrow y \quad (3)$$

The constraint parametrized

```
> pmin := -1 :
  pmax :=  1 :
```

```
> cc := unapply( [x, solve( hc(x,y), y )], x ) ;
```

$$cc := x \to [x, x^2] \quad (4)$$

```
> cs := unapply( [x, solve( hs(x,y), y )], x ) ;
```

$$cs := x \to [x, 0] \quad (5)$$

Compute the gradient at (0,0)

```
> gradHc := subs(x=0,y=0,<diff(hc(x,y),x)|diff(hc(x,y),y)>) ;
```

$$gradHc := \begin{bmatrix} 0 & 1 \end{bmatrix} \quad (6)$$

```
> gradHs := subs(x=0,y=0,<diff(hs(x,y),x)|diff(hs(x,y),y)>) ;
```

$$gradHs := \begin{bmatrix} 0 & 1 \end{bmatrix} \quad (7)$$

The lagrangian

```
> Ls := f(x,y) + lambda * hs(x,y) ;
  Lc := f(x,y) + lambda * hc(x,y) ;
```

$$Ls := 3 + y - \frac{1}{2} x^2 + \lambda y$$

$$Lc := 3 + y - \frac{1}{2} x^2 + \lambda \left(y - x^2\right) \quad (8)$$

The gradient of Lagrangian

```
> gradLs := [diff(Ls,x),diff(Ls,y),diff(Ls,lambda)] : <%> ;
  gradLc := [diff(Lc,x),diff(Lc,y),diff(Lc,lambda)] : <%> ;
```

$$\begin{bmatrix} -x \\ 1 + \lambda \\ y \end{bmatrix}$$

$$\begin{bmatrix} -x - 2 \lambda x \\ 1 + \lambda \\ y - x^2 \end{bmatrix} \quad (9)$$

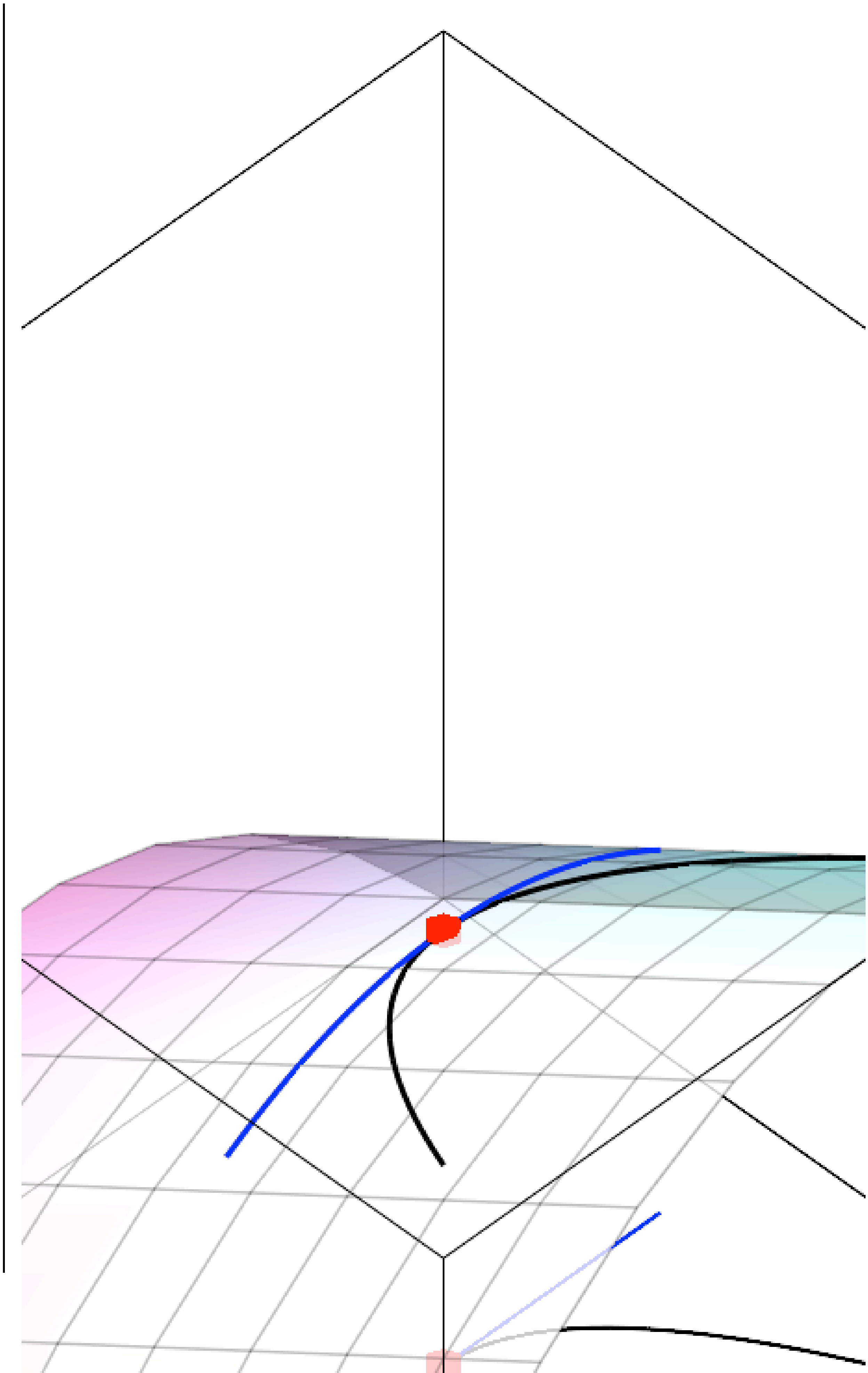
```
> SOLs := op(solve( gradLs, [x,y,lambda]));
  SOLc := op(solve( gradLc, [x,y,lambda]));
```

$$SOLs := [x = 0, y = 0, \lambda = -1]$$

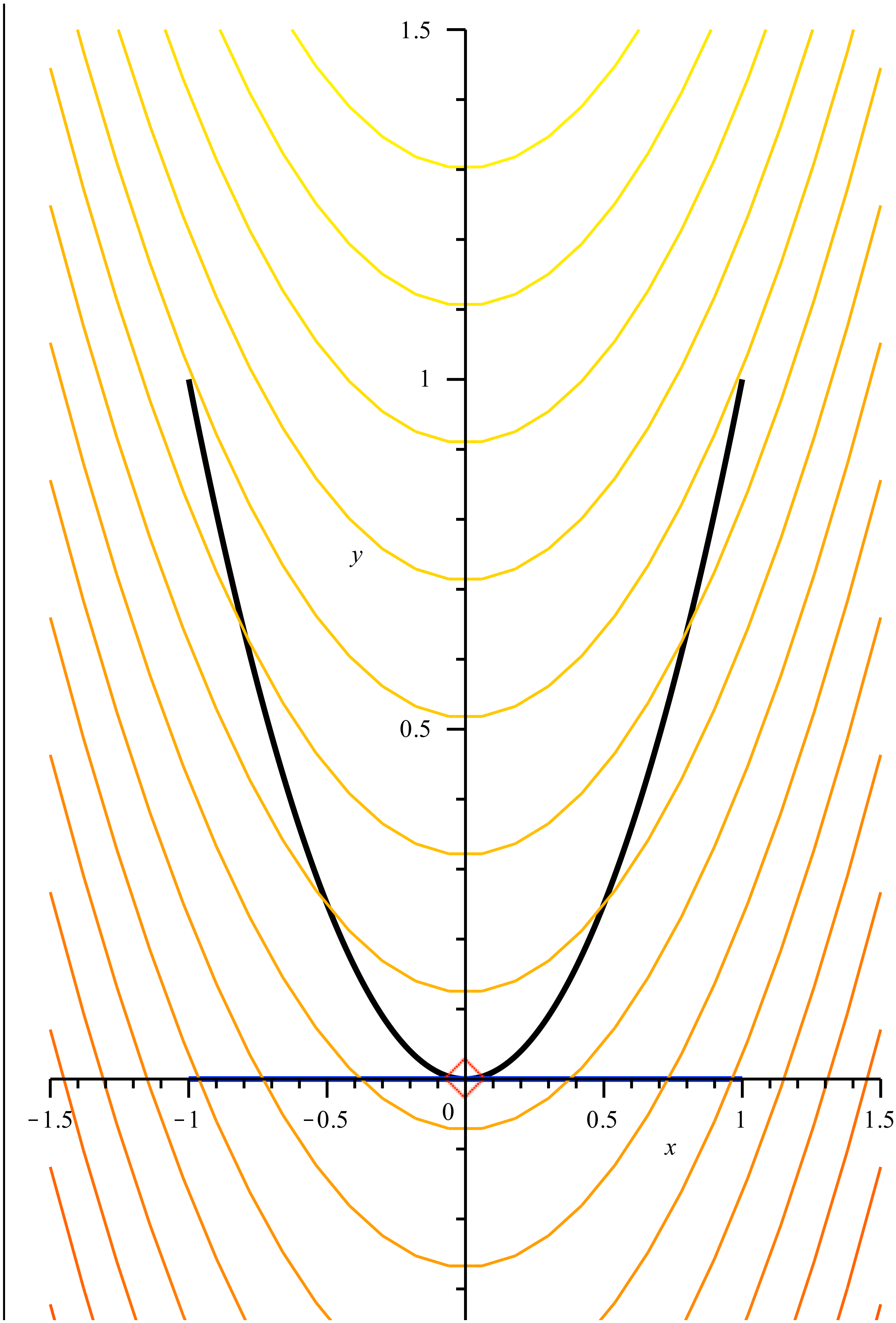
$$SOLc := [x = 0, y = 0, \lambda = -1] \quad (10)$$

```
> P0 := plot3d( [0], x=-1.5..1.5, y=-1.5..1.5, style=patch,axes=
  boxed, color=RGB(0.9,0.9,0.9), grid=[2,2] ):
  P1 := plot3d( f(x,y), x=-2..2, y=-2..2, style=patch,axes=boxed,
  grid=[10,10],transparency=0.5 ):
  P2 := spacecurve( [ op(cc(theta)),f(op(cc(theta)))], theta=pmin..
  pmax,thickness=3,color=black):
  P3 := spacecurve( [ op(cc(theta)),0], theta=pmin..pmax,thickness=2,
  color=black):
  P4 := spacecurve( [ op(cs(t)),f(op(cs(t)))], t=pmin..pmax,
  thickness=3,color=blue):
```

```
P5 := spacecurve( [ op(cs(t)),0], t=pmin..pmax,thickness=2,color=
blue):
P6 := pointplot3d( {[0,0,0],[0,0,f(0,0)]}, symbolsize=20, color=red
):
P7 := arrow([0,0,0], [gradH0[1]/2, gradH0[2]/2,0],0.2,0.4,0.1,
color = green):
> display(P||(1..7)) ;
```

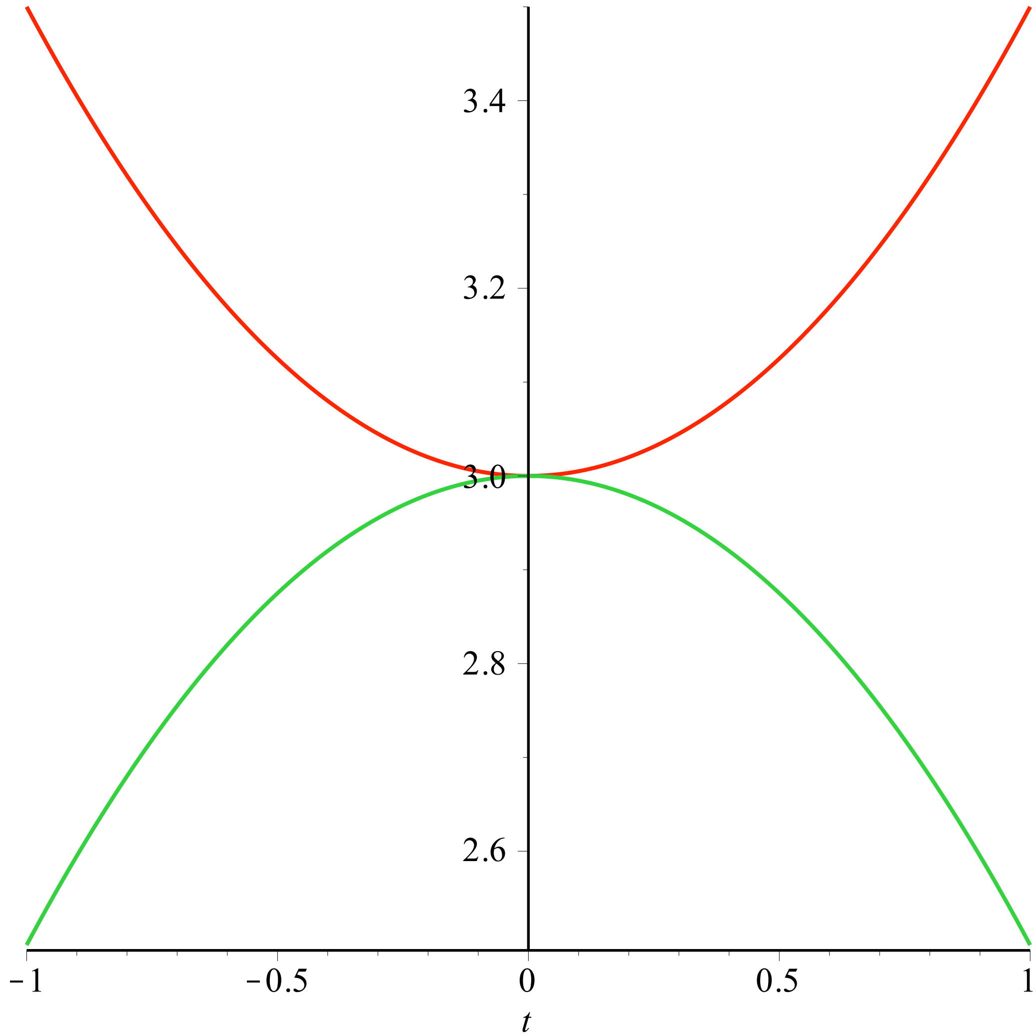


```
> P1 := plot( [ op(cc(theta)), theta=pmin..pmax],thickness=3,color=
  black):
  P2 := plot( [ op(cs(t)),t=pmin..pmax],thickness=3,color=blue):
  P3 := pointplot( {[0,0]}, symbolsize=20, color=red ):
  P4 := arrow([0,0], [gradH0[1]/2, gradH0[2]/2],0.2,0.4,0.1, color =
  green):
  P5 := contourplot(f(x,y),x=-1.5..1.5, y=-1.5..1.5,contours=20):
> display(P||(1..5)) ;
```



1.5
1
y
0.5
0
−1.5
−1
−0.5
0.5
1
1.5
x

```
> plot( [f(op(cc(t))), f(op(cs(t)))], t=pmin..pmax) ;
```




```
> Hessf := <<D[1,1](f)(x,y),D[1,2](f)(x,y)>|
            <D[2,1](f)(x,y),D[2,2](f)(x,y)>> ;
```

$$Hessf := \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix} \quad \textbf{(11)}$$

```
> HessHc := <<D[1,1](hc)(x,y),D[1,2](hc)(x,y)>|
             <D[2,1](hc)(x,y),D[2,2](hc)(x,y)>> ;
  HessHs := <<D[1,1](hs)(x,y),D[1,2](hs)(x,y)>|
             <D[2,1](hs)(x,y),D[2,2](hs)(x,y)>> ;
```

$$HessHc := \begin{bmatrix} -2 & 0 \\ 0 & 0 \end{bmatrix}$$

$$HessHs := \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \textbf{(12)}$$

```
> HessLc := subs( SOLc, Hessf + lambda * HessHc ) ;
  HessLs := subs( SOLs, Hessf + lambda * HessHs ) ;
```

$$HessLc := \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$HessLs := \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix} \quad (13)$$

FInd the kernel of gradH0

```
> KMATc := op(NullSpace(<gradHc>)) ;
  KMATs := op(NullSpace(<gradHs>)) ;
```

$$KMATc := \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$KMATs := \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (14)$$

Project the hessian in the kernel (with multiplier)

```
> Transpose(KMATc).HessLc.KMATc ;
  Transpose(KMATs).HessLs.KMATs ;
```

$$1$$

$$-1 \quad (15)$$

Project the hessian in the kernel

```
> Transpose(KMATc).Hessf.KMATc ;
  Transpose(KMATs).Hessf.KMATs ;
```

$$-1$$

$$-1 \quad (16)$$