

1. Parametrisierungen der beiden Kurven aufschreiben

```
[ w1:=( )-->matrix([])
[ w2:=( )-->matrix([])
[ plot(
  plot::Curve3d(w1( ), ),
  plot::Curve3d(w2( ), )
)
```

2. Zielfunktion aufstellen

$(x_1-x_2)^2+(y_1-y_2)^2+(z_1-z_2)^2$ Wobei x, y, z für die jeweilige Koordinate in der Kurve stehen

```
[ F:=( )-->(norm(w1( )-w2( ), 2))^2
```

3. Gradient und Jakobimatrix berechnen

```
[ gradF:=( )-->linalg::gradient(F( ), [ ]):
[ JgradF:=( )-->linalg::jacobian(gradF( ), [ ]):
```

4. Startwert bestimmen

```
[ X:=matrix([ ])
```

5. Newtonverfahren durchführen

```
[ for k from 1 to 10 do
  d:=float(linalg::matlinsolve(JgradF(X[i]$i=1.. ),-gradF(X[i]$i=1.. ))):
  X:=d+X:
  print(norm(d,2));
end_for;
[ X;
```

6. Plot zur Kontrolle

```
[ plot(
  plot::Curve3d(w1( ), ),
  plot::Curve3d(w2( ), ),
  plot::Line3d(w1( ), w2( )),
  Scaling = Constrained
)
```

7. Abstand bestimmen

```
[ float(norm(w1( )-w2( ), 2))
```

```
[
```