Inverse functions in Maple

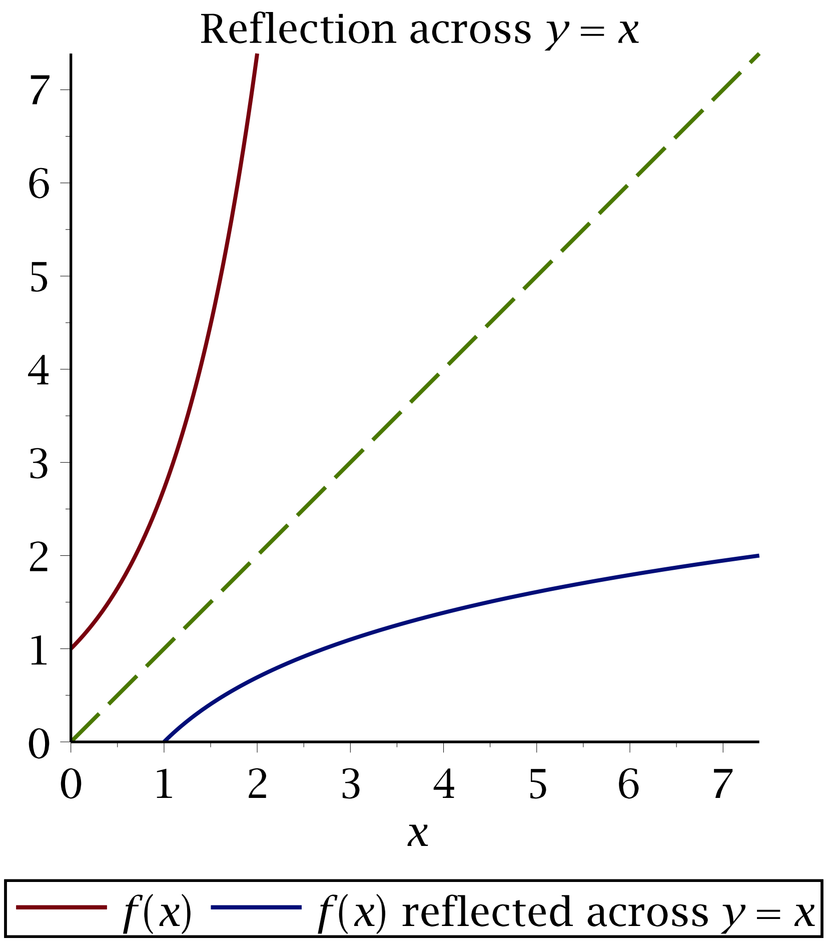
Inverse functions can be found and analysed using commands in the **Student**[**Calculus1**] package :

```
with(Student[Calculus1]);
```

```
[AntiderivativePlot, AntiderivativeTutor, ApproximateInt, ApproximateIntTutor,        (1)
    ArcLength, ArcLengthTutor, Asymptotes, Clear, CriticalPoints,
    CurveAnalysisTutor, DerivativePlot, DerivativeTutor, DiffTutor,
    ExtremePoints, FunctionAverage, FunctionAverageTutor, FunctionChart,
    FunctionPlot, GetMessage, GetNumProblems, GetProblem, Hint,
    InflectionPoints, IntTutor, Integrand, InversePlot, InverseTutor, LimitTutor,
    MeanValueTheorem, MeanValueTheoremTutor, NewtonQuotient,
    NewtonsMethod, NewtonsMethodTutor, PointInterpolation, RiemannSum,
    RollesTheorem, Roots, Rule, Show, ShowIncomplete, ShowSolution,
    ShowSteps, Summand, SurfaceOfRevolution, SurfaceOfRevolutionTutor,
    Tangent, TangentSecantTutor, TangentTutor, TaylorApproximation,
    TaylorApproximationTutor, Understand, Undo, VolumeOfRevolution,
    VolumeOfRevolutionTutor, WhatProblem]
```

The **InversePlot** command plots the function and its possible inverse on a given interval . For example, the expontial function on the interval [0, 2] :

```
InversePlot(exp(x), x = 0 ..2);
```



A graph of $f(x) = e^{x}$. The line $y = x$. The reflection of $f(x)$ across the line $y = x$.

To see that the blue line is indeed the graph of the logarithmic function, use the **InverseTutor** command . This will open a small window where you can see the formula **for** the inverse **and** also specify other **options.**

```
InverseTutor(exp(x), x = 0 ..2);
```

In case the inverse is not well defined, Maple answers with more than one alternative, as is the case with x^2 :

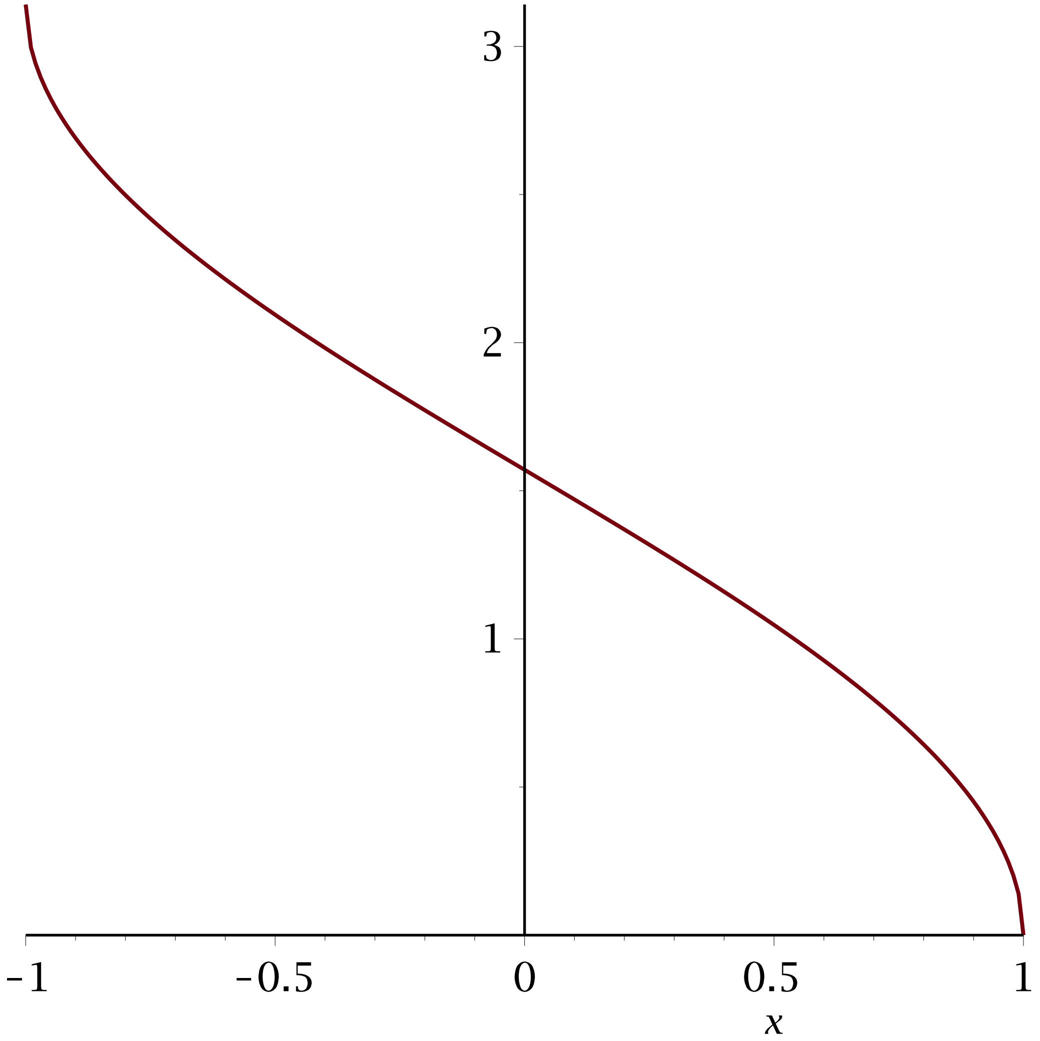
```
InverseTutor(x^2, x = -1 ..1);
```

Note that the inverse is only defined where the function is one – **to**–one, **for** example cos(x) on the interval [0, Pi] :

```
InverseTutor(cos(x), x = 0 ..Pi);
```

The inverse of **cos**, called **arccos**, is defined on the interval [-1, 1] and can also be visualised directly using the **plot** command :

```
plot(arccos(x), x =-1 ..1);
```


3
2
1
-1
-0.5
0
0.5
1
x