```
sol = NDSolve[{x'[t] == 2 x[t] + y[t] * y[t] - 1,
    y'[t] == 6 x[t] - y[t] * y[t] + 1, x[0] == 1, y[0] == 1}, {x[t], y[t]}, {t, 0, tend}]
```

$$\left\{\left\{x[t] \rightarrow \text{InterpolatingFunction}\left[\boxed{\begin{array}{ll} \text{Domain:} & \{\{0., 1.\}\} \\ \text{Output:} & \text{scalar} \end{array}}\right][t]\right.\right.,$$

$$\left.\left. y[t] \rightarrow \text{InterpolatingFunction}\left[\boxed{\begin{array}{ll} \text{Domain:} & \{\{0., 1.\}\} \\ \text{Output:} & \text{scalar} \end{array}}\right][t]\right\}\right\}$$

```
plotX = Plot[x[t] /. sol, {t, 0, tend}, PlotStyle → Black];
plotY = Plot[y[t] /. sol, {t, 0, tend}, PlotStyle → Black];
h = 0.05; t0 = 0; tend = 1;
grid = Table[i, {i, t0, tend, h}];
```

```
runge = Import[
    "/Users/ivandybko/Projects/Numerical_methods/Mathematical physics/Lab1
      (Numerical methods for solving ordinary differential
      equations)/data/test1/runge.txt", "Table"];
expeuler = Import[
    "/Users/ivandybko/Projects/Numerical_methods/Mathematical physics/Lab1
      (Numerical methods for solving ordinary differential
      equations)/data/test1/explicit_euler.txt", "Table"];
impeuler = Import[
    "/Users/ivandybko/Projects/Numerical_methods/Mathematical physics/Lab1
      (Numerical methods for solving ordinary differential
      equations)/data/test1/implicit_euler.txt", "Table"];
symmetric = Import[
    "/Users/ivandybko/Projects/Numerical_methods/Mathematical physics/Lab1
      (Numerical methods for solving ordinary differential
      equations)/data/test1/symmetric.txt", "Table"];
adams = Import[
    "/Users/ivandybko/Projects/Numerical_methods/Mathematical physics/Lab1
      (Numerical methods for solving ordinary differential
      equations)/data/test1/adams_bashforth.txt", "Table"];
predictor = Import[
    "/Users/ivandybko/Projects/Numerical_methods/Mathematical physics/Lab1
      (Numerical methods for solving ordinary differential
      equations)/data/test1/adams_bashforth_with_predictor_corrector.txt",
    "Table"];
```

In[1598]:=
```
plotRungeX = ListPlot[Transpose[{grid, runge⟦All, 1⟧}], PlotStyle → Red];
plotRungeY = ListPlot[Transpose[{grid, runge⟦All, 2⟧}], PlotStyle → Red];
plotExpeulerX = ListPlot[Transpose[{grid, expeuler⟦All, 1⟧}], PlotStyle → Red];
plotExpeulerY = ListPlot[Transpose[{grid, expeuler⟦All, 2⟧}], PlotStyle → Red];
plotImpeulerX = ListPlot[Transpose[{grid, impeuler⟦All, 1⟧}], PlotStyle → Red];
plotImpeulerY = ListPlot[Transpose[{grid, impeuler⟦All, 2⟧}], PlotStyle → Red];
plotSymmetricX = ListPlot[Transpose[{grid, symmetric⟦All, 1⟧}], PlotStyle → Red];
plotSymmetricY =
  ListPlot[Transpose[{grid, symmetric⟦All, 2⟧}], PlotStyle → Red];
plotAdamsX = ListPlot[Transpose[{grid, adams⟦All, 1⟧}], PlotStyle → Red];
plotAdamsY = ListPlot[Transpose[{grid, adams⟦All, 2⟧}], PlotStyle → Red];
plotPredictorX = ListPlot[Transpose[{grid, predictor⟦All, 1⟧}], PlotStyle → Red];
plotPredictorY =
  ListPlot[Transpose[{grid, predictor⟦All, 2⟧}], PlotStyle → Red];
```

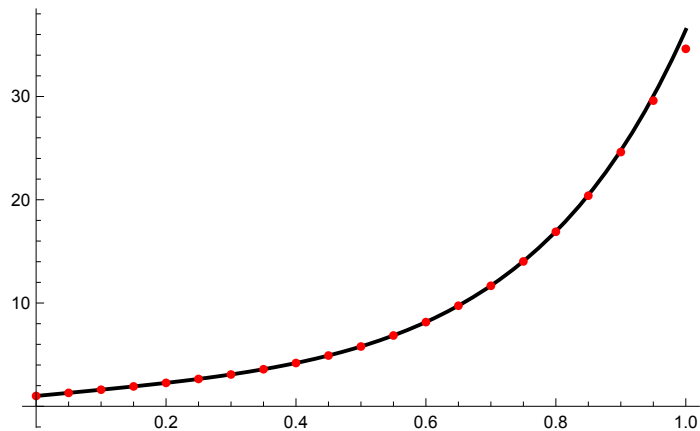In[1604]:=
```
Show[plotX, plotRungeX, PlotRange → All]
```

Out[1604]=
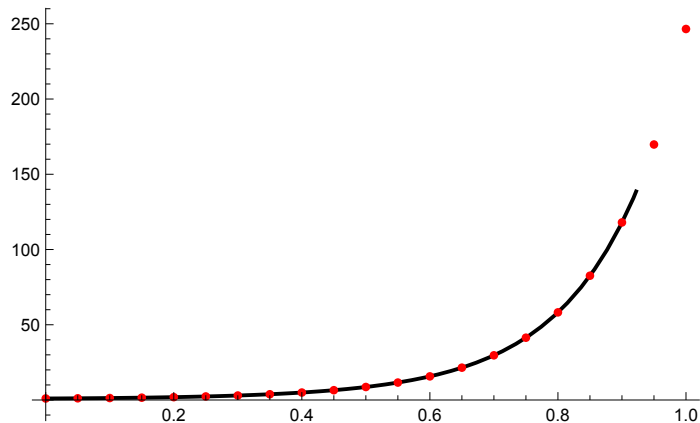
In[1605]:=
```
Show[plotY, plotRungeY, PlotRange → All]
```
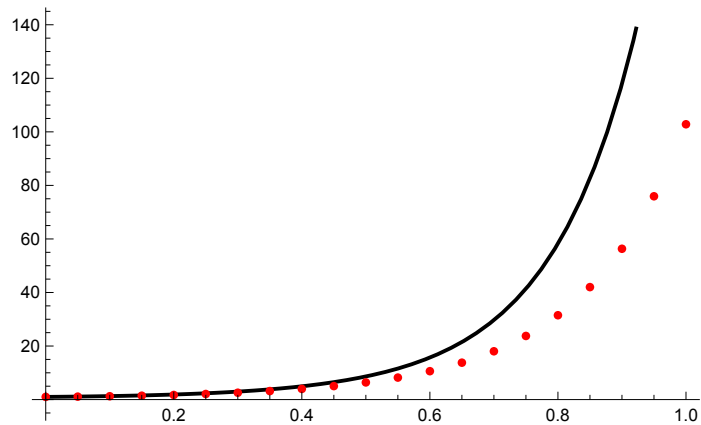
Out[1605]=

In[1606]:=

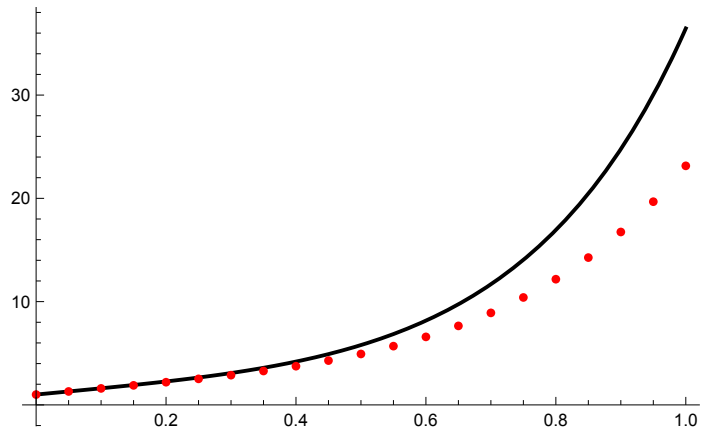**Show[plotX, plotExpeulerX, PlotRange → All]**

Out[1606]=

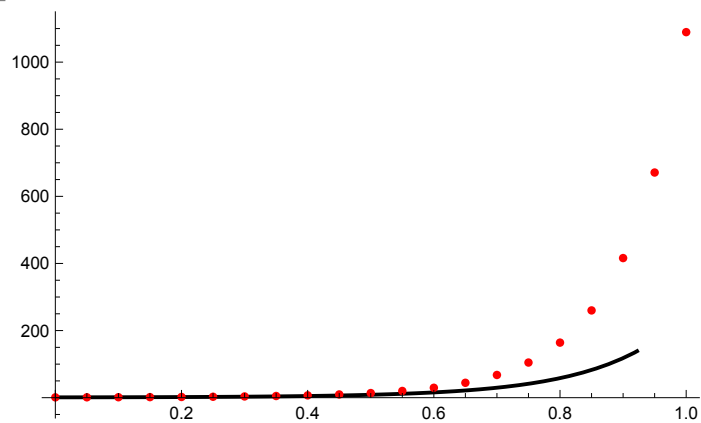In[1607]:=

Show[plotY, plotExpeulerY, PlotRange → All]

Out[1607]=

In[1608]:=

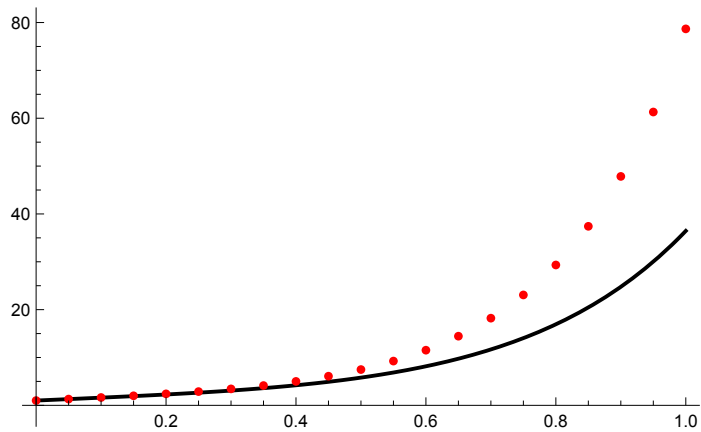**Show[plotX, plotImpeulerX, PlotRange → All]**

Out[1608]=

In[1609]:=
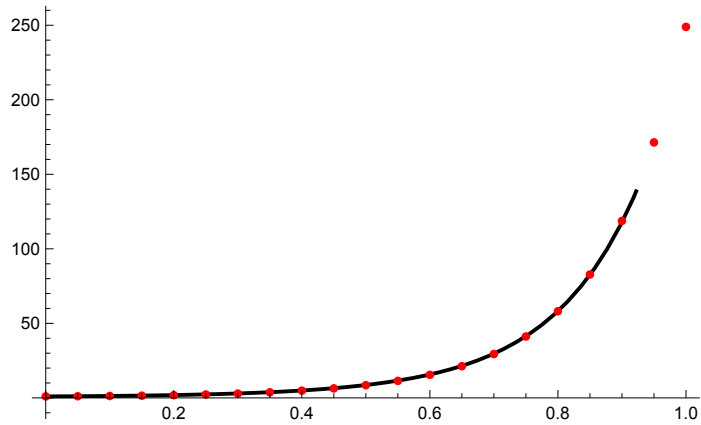
Show[plotY, plotImpeulerY, PlotRange → All]

Out[1609]=

In[1610]:=

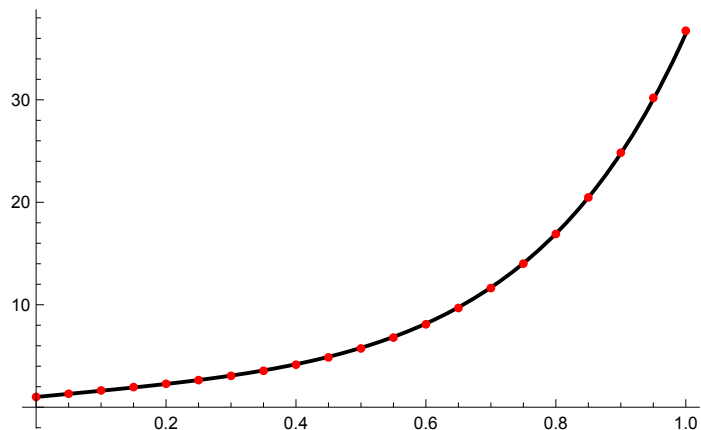**Show[plotX, plotSymmetricX, PlotRange → All]**

Out[1610]=

In[1611]:=

Show[plotY, plotSymmetricY, PlotRange → All]
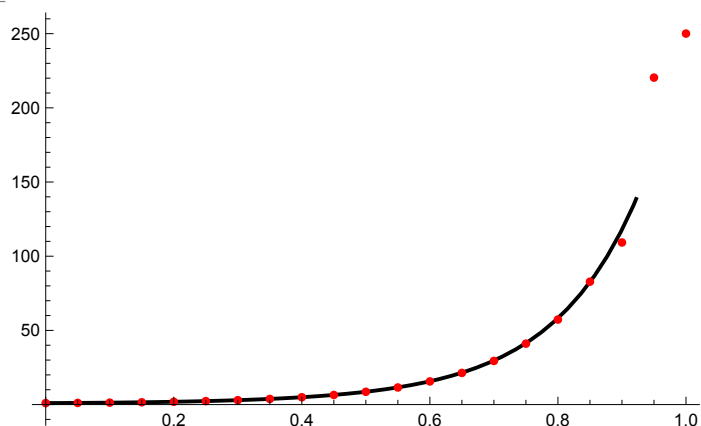
Out[1611]=

In[1612]:=
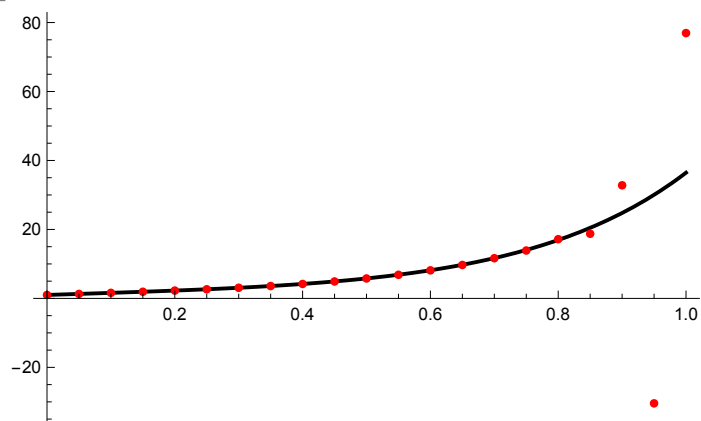
Show[plotX, plotAdamsX, PlotRange → All]

Out[1612]=

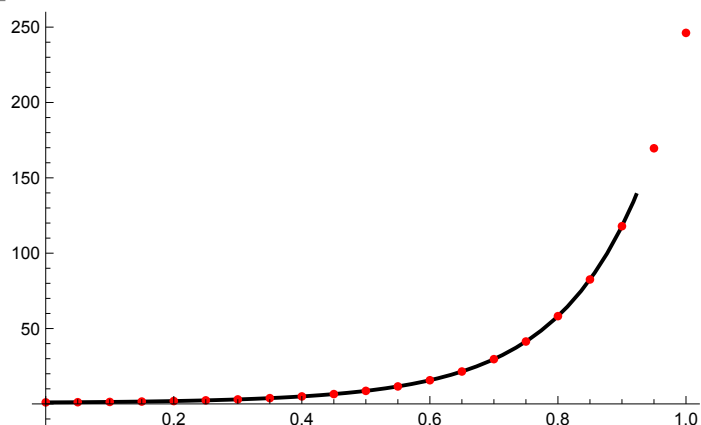

In[1613]:=

Show[plotY, plotAdamsY, PlotRange → All]

Out[1613]=
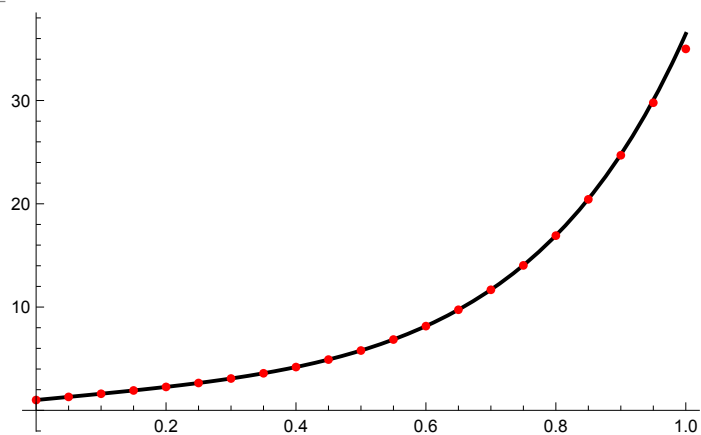

In[1614]:=

Show[plotX, plotPredictorX, PlotRange → All]

Out[1614]=

In[1615]:=

Show[plotY, plotPredictorY]

Out[1615]=

In[1616]:=